



A University of Sussex PhD thesis

Available online via Sussex Research Online:

<http://sro.sussex.ac.uk/>

This thesis is protected by copyright which belongs to the author.

This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the Author

The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the Author

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given

Please visit Sussex Research Online for more information and further details

The quantum chemical physics of few-particle atoms and molecules



Adam Baskerville

Supervisor: Dr Hazel Cox

Department of Chemistry
University of Sussex

This dissertation is submitted for the degree of
Doctor of Philosophy

July 2018



Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Adam Baskerville
July 2018



Acknowledgements

First and foremost I would like to thank my supervisor, Dr Hazel Cox for the unwavering support and guidance she has given me throughout my PhD studies; and showing me, by her example, what it means to be a good scientist. I would also like to thank Professor Anthony Mccaffery for many insightful conversations and his support when preparing for presentations. Additionally, I would like to thank Dr John Turner for coaching me when preparing for conference talks.

Special thanks also go to the Cox lab group members, past and present, for making the office a fun and supportive environment to work in: Dr Andrew King, Dr Gavin Roffe, Dr Sarote Boonseng (Pop), Robert Ziolek, Msugh Targema. I wish you all every success in your future endeavours.

I would also like to extend my deepest gratitude to Charlotte for her love and encouragement throughout my studies.

Last, but by no means least I would like to thank my family including my Nan and especially my Mum who raised me with a love and admiration of science and maths.



Thesis Output

Publications:

- A. L. Baskerville, A. W. King and H. Cox. Quantum effects of nuclear motion in three particle diatomic ions, *Phy. Rev. A*. **94**. 042512. 2016
- A. W. King, A. L. Baskerville and H. Cox. Hartree–Fock implementation using a Laguerre-based wave function for the ground state and correlation energies of two-electron atoms, *Phil. Trans. R. Soc. A*. **376**. 2017
- H. Cox and A. L. Baskerville. The Series Solution Method in Quantum Chemistry for Three-Particle Systems, *Adv. Quant. Chem.* **77**. 2018
- A. L. Baskerville, A. W. King and H. Cox. Electron correlation: energies, densities and Coulomb holes, *Submitted for review*

Oral presentations:

- “*Excited S states of two electron atoms*” - Quantum Physics and Quantum Technology, Berlin, Germany, 2017
- “*Going beyond standard approximations in quantum chemistry*” - Young Modellers Forum (YMF), University of Greenwich, London, UK, 2017
- “*Going beyond standard approximations in quantum chemistry*” - School of Life Sciences research symposium, University of Sussex, Brighton, UK, 2017

Poster presentations:

- “*Quantum effects of particle motion in atomic and molecular three-particle systems*” - IX Congress of the International Society for Theoretical Chemical Physics (ISTCP), North Dakota, USA, 2016
- “*Intracule and centre of mass particle densities for atomic and molecular three - particle systems*” - International Meeting on Atomic and Molecular Physics and Chemistry (IMAMPC), Birmingham University, UK, 2015
- “*Quantum effects of particle motion in atomic and molecular three-particle systems*” - Life Sciences research symposium, University of Sussex, UK, 2016

Awards:

- 2017 Young Modellers Forum (YMF) talk prize
- 2016 Doctoral Overseas Grant award, The University of Sussex
- 2016 Institute of Physics funding award, Institute of Physics

Abstract

The many-electron Schrödinger equation for atoms and molecules still remains analytically insoluble after over 90 years of investigation. This has not deterred scientists from developing a large variety of elegant techniques and approximations to workaroud this issue and make many-particle quantum calculations computationally tractable. This thesis presents an all-particle treatment of three-particle systems which represent the simplest, most complex, many-particle systems including electron correlation and nuclear motion effects; meaning they provide a close-up view of fundamental particle interaction. Fully-Correlated (FC) energies and wavefunctions are calculated to high accuracy (mJ mol^{-1} or better for energies); and the central theme of this work is to use the wavefunctions to study fundamental quantum chemical physics.

Nuclear motion has not received the same attention as electronic structure theory and this complicated coupling of electron and nuclear motions is studied in this work with the use of intracule and centre of mass particle densities where it is found nuclear motion exhibits strong correlation.

A highly accurate Hartree-Fock implementation is presented which uses a Laguerre polynomial basis set. This method is used to accurately calculate electron correlation energies using the Löwdin definition and Coulomb holes by comparing with our FC data. Additionally the critical nuclear charge to bind two electrons within the HF methodology is calculated.

A modification to Pekeris' series solution method is implemented to accurately model excited states of three-particle systems, and adapted to include the effects of nuclear motion along with three Non-Linear variational Parameters (NLPs) to aid convergence. This implementation is shown to produce high accuracy results for singlet and triplet atomic excited S states and the critical nuclear charge to bind two electrons in both spin states is investigated.

Geometrical properties of three-particle systems are studied using a variety of particle densities and by determining the bound state stability at the lowest continuum threshold as a function of mass. This enables us to better ascertain what is meant when we define a system as an atom or a molecule.

Contents

List of Figures	ix
List of Tables	xi
Abbreviations	xiv
1 Introduction	1
1.1 Scope of this work	2
2 Theory	5
2.1 Atomic Units	5
2.2 Non-Relativistic, Time Independent Schrödinger Equation for Three- Particle Systems	6
2.2.1 Inter-particle and Perimetric Coordinates	6
2.2.2 Wavefunction form	7
2.2.3 Fermionic Exchange	8
2.2.4 Non-Linear Variational Parameters: Wavefunction forms	9
2.2.5 Series Solution Method	11
2.3 Eigenvalue Problem	13
2.4 Expectation Values: Variational Theorem	14
2.5 Virial Condition	16
2.6 Cusp Condition	16
2.7 Particle Densities	17
2.8 Born-Oppenheimer Approximation	19
2.9 Hartree-Fock Theory	20
2.10 Summary and Conclusions	22
3 Computer Programming	23
3.1 Introduction	23
3.2 Program map	24
3.3 Hardware description	25
3.3.1 Build environment	25
3.4 C++	26
3.4.1 The 3Body code	26
3.4.1.1 3Body Code Validation	35

3.4.2	Expectation Values	37
3.4.3	Numerical integration	38
3.5	Maple	44
3.5.1	Recursion Relations	44
3.5.2	Inter-Particle Cusps	49
3.5.3	Numerical Integration	52
3.5.4	Excited State Matrices	55
3.6	Python	61
3.6.1	Choice of Python	61
3.6.1.1	Language flexibility	61
3.6.1.2	Language Interoperability	61
3.6.2	Excited state code development	63
3.6.2.1	Loss of sparsity	63
3.6.2.2	Asymmetric numbering scheme	64
3.6.2.3	Ill conditioned eigenvalue problem	65
3.6.3	Processing matrices	67
3.6.3.1	Compiling matrix elements	68
3.6.3.2	Evaluating matrix elements	69
3.6.3.3	Generating less general matrices	70
3.6.3.4	Combination of cexprtk, eval and custom matrix creation	71
3.6.3.5	Using computer architecture - Multithreading	72
3.6.4	Calculation options	74
3.7	Bash	76
3.7.1	Main Shell Scripts	76
3.8	Summary and Conclusions	79
4	Nuclear motion in three-particle diatomic ions	80
4.1	Introduction	80
4.2	Method	81
4.2.1	Barycentric coordinates	81
4.2.2	New Coordinate System	83
4.3	Implementation	87
4.4	Results and Discussion	91
4.4.1	Comparison with Born-Oppenheimer approximation	91
4.4.2	Intracule Densities	95
4.4.3	Centre of Mass Particle Densities	98
4.4.3.1	Homonuclear systems	98
4.4.3.2	Heteronuclear systems	100
4.5	Summary and Conclusions	107
5	Electron correlation: energies, densities and Coulomb holes	108

5.1	Introduction	108
5.2	Method	110
5.2.1	Schrödinger equation and choice of coordinates	110
5.2.2	The Hartree-Fock Method	114
5.2.2.1	Implementation	116
5.2.3	Coulomb hole calculation	121
5.2.3.1	Implementation	122
5.3	Results and Discussion	124
5.3.1	Literature comparison	124
5.3.2	Energy Convergence	125
5.3.3	Wavefunction Quality: Bound State Properties	127
5.3.4	Electron Correlation Energies	129
5.3.5	Critical nuclear charge for binding under the Hartree-Fock approximation	130
5.4	Electron density and Coulomb holes	136
5.4.1	Helium atom and Lithium cation	136
5.4.2	Hydride anion and the critical nuclear charge system	138
5.4.3	Coulomb hole comparison	139
5.4.4	Long range behaviour: Search for additional Coulomb holes	140
5.5	Summary and Conclusions	141
6	Excited S states of two electron atoms	144
6.1	Introduction	144
6.2	Method	145
6.2.1	Relaxing $\alpha = \beta$ NLP constraint	146
6.2.2	Series solution	147
6.2.3	Implementation	149
6.2.3.1	Inclusion of nuclear motion	150
6.2.3.2	Inclusion of three NLPs, α, β, γ	151
6.2.4	Variational principle for excited states	152
6.3	Results and Discussion	153
6.3.1	Energy convergence	153
6.3.2	Effect of principle quantum number, n , on NLPs, α, β, γ	156
6.3.3	Effect of nuclear motion	158
6.3.4	Comparison to Ground State Method	160
6.3.4.1	Singlet state energies	162
6.3.4.2	Triplet state energies	163
6.3.5	Expectation values	164
6.3.6	Critical Nuclear Charge	167
6.3.6.1	Z_C : Singlet Ground State	169
6.3.6.2	Z_C : Excited Singlet and Triplet States	170
6.4	Summary and Conclusions	173

7	Atomic and molecular structure from an all-particle wavefunction	175
7.1	Introduction	175
7.2	Method	177
7.2.1	Angular Densities	178
7.2.2	Lower bound to stability	179
7.3	Results and Discussion	181
7.3.1	Emergence of Molecular Structure?	181
7.3.1.1	Radial densities	181
7.3.1.2	Angular densities	185
7.3.2	Gradient of COM particle density distributions	187
7.3.3	Classical structure from quantum mechanics	191
7.4	Summary and Conclusions	194
8	Thesis Summary and Conclusions	196
	Bibliography	199
	Appendix A	214
A.1	Ground state Run file	214
A.2	Excited state Run file	216
	Appendix B	222
B.1	Grand Central Dispatch - Apache License	222
B.2	MPFRC++ - General Public License	222
B.3	Doublepresso - MIT License Copyright	223
B.4	ASMJit - Zlib License	224
B.5	Numerical Recipes Personal Single-User License	224
B.6	Lapack - Modified BSD License	225
B.7	Boost Software License	226
B.8	GALib - MIT License	227
B.9	Common Public License, version 1.0	228
	Appendix C	234
C.1	HF two-electron integral derivation	234

List of Figures

2.1	Inter-particle coordinates and perimetric coordinates	7
2.2	laguerre polynomial functions	8

2.3	Particle densities	18
3.1	Programming map	24
3.2	CPU times for C++ and python ground state codes vs. basis set size .	37
3.3	Python wrapper vs. C implementation of the OpenBlas dot product function	62
3.4	Sparsity comparison between \mathbf{H} for ground state and excited state methodologies	64
3.5	Multiprocessing of matrix elements in Python	74
4.1	Representation of Barycentric coordinates	82
4.2	Centre of mass coordinate system using combination of inter-particle coordinates and Barycentric coordinates	83
4.3	Three-particle system relative to arbitrary origin, O	84
4.4	Intracule densities $h(r)$ for electronic systems and muonic systems .	95
4.5	Density plots of $\rho_{c,a_i}(s)$ for $s = (s_x, s_y, 0)$ in the case of homonuclear systems where $a_1^+ = a_2^+$	98
4.6	Centre of mass particle densities, $\rho_{c,a_i}(s)$, for electronic and muonic homonuclear diatomic systems	99
4.7	Density plots of $\rho_{c,a_i}(s)$ for systems where $a_1^+ \neq a_2^+$	101
4.8	centre of mass particle densities, $\rho_{c,a_i}(s)$, for heteronuclear diatomic ions	102
4.9	Localisation of the deuteron radial centre of mass particle density distribution in $p^+d^+e^-$ and $d^+t^+e^-$	104
4.10	Radial centre of mass particle densities for $d^+t^+e^-$ highlighting ratio of probability maxima and FWHM	105
5.1	Expectation values, $\langle r_1 \rangle$ and $\langle r_{12} \rangle$ vs. nuclear charge for low charge HF	134
5.2	Coulomb holes for lithium cation and helium atom	136
5.3	Coulomb holes for (a) hydride and (b) critical nuclear charge	138
5.4	Z-scaled Coulomb holes.	139
5.5	Long-range behaviour of the intracule distribution functions for (a) He and (b) H^-	141
5.6	Long-range behaviour of the intracule distribution functions $D_i(r)$, $i = FC$ or HF , and the Coulomb hole function, Δ , for the hydride ion, H^-	141
6.1	Effect of principle quantum number, n on the values of the NLPs, compared against values used by Pekeris	157

6.2	Energy convergence comparison for the first 9 excited states of singlet helium with a clamped nucleus between the ground state 2 NLP (α, γ) method and the excited state 3 NLP (α, β, γ) method	162
6.3	Energy convergence comparison for the first 9 excited states of triplet helium with a clamped nucleus between the ground state 2 NLP (α, γ) method and the excited state 3 NLP (α, β, γ) method	163
7.1	Eckart coordinate system	178
7.2	$g(A_3)$ vs. A_3 for symmetric systems	181
7.3	Centre of mass particle density distributions for (a) H^- , (b) Ps^- , (c) H_2^+	182
7.4	Centre of mass particle density distributions for arbitrary mass ratios, $m_{a_1}/m_{a_3} \in [0.4, 2.0]$	184
7.5	Zoomed in centre of mass particle density distributions for arbitrary mass ratios, $m_1/m_3 \in [0.4, 0.8]$	185
7.6	Angular particle density distributions for (a) H^- , (b) Ps^- , (c) H_2^+ . .	186
7.7	Angular particle density distributions for arbitrary mass ratios, $m_{a_1}/m_{a_3} \in [0.4, 2.0]$	187
7.8	Distance from uniquely charged particle, a_3^- to the centre of mass, c .	189
7.9	Centre of mass particle density distributions for arbitrary mass ratios, $m_{a_1}/m_{a_3} \in [0.5, 2.0]$ measured from particle a_3	190
7.10	Most probable and average features of H^- , Ps^- and H_2^+	193

List of Tables

2.1	Atomic and muon atomic units	5
2.2	ASYM, SYM and ANTISYM numbering schemes	13
3.1	Double vs. quadruple precision	28
3.2	Optimisation algorithm comparison for He	33
3.3	Optimisation algorithm comparison for H_2^+	33
3.4	Optimisation using fork method for He and H_2^+	34
3.5	Verification of the 3Body code for atomic systems	36
3.6	Verification of the 3Body code for molecular systems	36
3.7	Expansion method vs. gradient method for cusp calculations	52

3.8	Comparison of byte size for the lower triangular matrix representing \mathbf{H} using the ground state and excited state methodologies	65
3.9	Math parser comparison	68
3.10	Byte size for the lower triangular Hamiltonian matrix for atoms in the excited state method where $m_1 = m_2 = 1, Z_1 = Z_2 = -1$	71
4.1	Energy, dissociation energy and bond-length for electronic and muonic three-particle systems	94
4.2	Maximum r and maximum $h(r)$ for intracule plots	97
4.3	Features of the radial center of mass particle density distributions . .	106
5.1	Coefficients in the recursion relation	117
5.2	Hartree-Fock energy comparison with literature	125
5.3	Comparison of helium energy with increasing basis size with numerical HF	126
5.4	Effect of NLP A on convergence of HF helium energy	126
5.5	Effects of electron correlation on bound state properties of Z_C , H^- , He and Li^+	128
5.6	Electron correlation energies	130
5.7	Convergence of the Hartree Fock critical nuclear charge	131
5.8	Critical nuclear charge correlation energies	132
5.9	Expectation value convergence for low, nuclear charge Z as a function of basis set size m	133
5.10	Hartree Fock energies for low nuclear charge, $Z < 1$	135
5.11	Roots, area and minima of the Coulomb holes in the helium-like ions	137
6.1	Energy convergence data for the excited $2^1S, 4^1S, 6^1S, 8^1S, 10^1S$ and 12^1S states of helium using three NLPs, α, β, γ with clamped nucleus and nucleus in motion	154
6.2	Energy convergence data for the excited $2^3S, 4^3S, 6^3S, 8^3S, 10^3S$ states of helium using three NLPs, α, β, γ with clamped nucleus and nucleus in motion	155
6.3	Effects of nuclear motion on excited singlet states of helium	159
6.4	Effects of nuclear motion on excited triplet states of helium	159
6.5	Energy for the 2^1S and 10^1S of clamped nucleus helium calculated using the ground state with two NLPs (α, γ) and the excited state method with three NLPs, α, β, γ	164
6.6	Expectation values for n^1S states, $n = 2, 4, 6, 8, 10$ of helium.	166

6.7	Inner product of the excited state 10^1S with lower lying excited state for a single diagonalisation and when each state is optimised individually.	167
6.8	Comparison of the critical nuclear charge, Z_C calculated using the ground state method and the excited state method	170
6.9	Critical nuclear charge, Z_C , for the 2^1S and 2^3S state of helium with clamped nucleus.	171
6.10	Critical nuclear charge, Z_C , for the 2^3S state of helium with clamped nucleus calculated using the ground state method with two NLPs α, γ .	172
6.11	Critical nuclear charge, Z_C , for the 2^1S state of helium with clamped nucleus calculated using the excited state method with two NLPs α, β .	173
6.12	Critical nuclear charge, Z_C , for the 2^1S state of helium with clamped nucleus calculated using the excited state method with three NLPs α, β, γ	173
7.1	Energy and virial comparison with Mátyus <i>et al.</i>	182
7.2	Gradient of the centre of mass particle density distributions at the centre of mass position	188
7.3	Expectation values of θ_1, θ_2 and θ_{12} for atoms and molecules	192



Abbreviations

BASH	Bourne Again SHell
BOBYQA	Bound Optimisation BY Quadratic Approximation
BO	Born Oppenheimer Approximation
CC	Coupled Cluster
CI	Configuration Interaction
COM	Centre Of Mass
DFT	Density Functional Theory
DIIS	Direct Inversion of Iterative Space
FC	Fully Correlated
FWHM	Full Width at Half Maximum
GUI	Graphical User Interface
HF	Hartree-Fock
IDE	Integrated Development Environment
JIT	Just In Time
MR3	Multiple relatively robust representations
NLP	Non-linear variational parameter
NROpt	Numerical Recipes Optimisation
OO	Object Oriented
SCF	Self Consistent Field
ZPE	Zero Point Energy
pd	particle density
sv	save vector

CHAPTER 1



Introduction

The three-body problem has fascinated scientists for centuries with both its classical and quantum mechanical analogues. In 1687, Newton showed that his laws of gravitation and motion could readily predict the orbits of two interacting celestial bodies but when applied to three or more, they were unable to correctly solve the system^[1]. A general solution to the classical three-body problem was sought up until the late nineteenth century with no success and in 1887, Bruns, with subsequent corrections by Poincaré, presented an impossibility proof claiming no general analytical solution exists for the n -body problem ($n > 2$)^[2]. This led to an unfortunate stifling in research into the field even though these papers were misleading. What they had shown was that no closed form solutions exist that are producible from a particular method of integration; and whilst this is important and certainly advanced our understanding of differential equations it propelled the idea that *no analytical solutions exist for the n -body problem*. In 1912, Sundman showed that an exact series solution exists for the three-body problem^[3] which was generalised to the many-body problem by Wang in 1991^[4]. In practice, these series solutions result in slow convergence and are thus computationally infeasible for real world applications^[5].

With the advent of quantum mechanics in the early twentieth century, attention quickly turned to the quantum mechanical analogue of the three-body problem, with the first successful quantum mechanical treatment of the hydrogen molecular ion (H_2^+) by Burrau in 1927^[6], just one year after the breakthrough of wave mechanics by Erwin Schrödinger. Hylleraas was another of the early pioneers, whose work on the helium atom in 1929 included the inter-electronic distance (r_{12}) as an expansion term within the coordinate set^[7]. Bartlett later showed that Hylleraas had not presented a formal solution to the Schrödinger equation^[8], and proposed a correction with the inclusion of logarithmic terms in the expansion^[9].

In 1958 Fock reported the formal solution to the helium atom with a clamped nucleus^[10] which was proven later to be point-wise convergent for S-states^[11]; however determining the exact form of these expansion terms has proved extremely challenging. In the same year, Pekeris^[12] presented the most accurate calculations at the time for the ground state energies of two-electron atoms making use of the recursion

properties of Laguerre polynomials along with perimetric coordinates introduced by James and Coolidge^[13], all without using logarithmic terms.

Pekeris’ original method was only valid for atoms with a clamped nucleus but subsequent modifications by various authors^[14–18] have enhanced the method to explicitly include nuclear motion along with the addition of a number of non-linear variational parameters in order to improve the rate of convergence of the method. These modifications also allowed for the study of three-particle diatomic systems^[19–23].

In the present day, several authors have presented very high accuracy energy calculations for three-particle systems^[24–30] with Nakashima *et al.* reporting 28 converged significant figures for triplet helium with nucleus in motion^[28]. The significance of such a result is debatable considering the nuclear mass of helium is defined to 11 significant figures under the 2006 CODATA used by the author^[31]. Similar accuracies have also been obtained for the energy of the hydrogen molecular ion^[30].

Three-body systems still remain the “quantum laboratory” for studying the complex dynamics between sub-atomic particles interacting via Coulomb potentials, and hence make up the core of the work to follow.

1.1 Scope of this work

With the wealth of investigation into the quantum mechanical three-body problem over the last 9 decades, it is necessary to justify the work conducted in this thesis and to state its importance.

Applying quantum mechanics to many-body systems unveils the world of quantum chemistry; whose aim is to use quantum mechanics to accurately calculate and predict the behaviour of chemical systems and to investigate phenomena discoverable from experiment. In order to do this, various approximations must be made to make calculations more computationally tractable, with two key ones being the Born-Oppenheimer approximation and the mean field approach of Hartree-Fock theory which forms the foundation of computational chemistry codes such as Gaussian^[32] and ADF^[33]. Effort is then required to recover the correlation energy and the effects of nuclear motion without losing the computational advantage gained by the assumptions. Performing high accuracy computations without approximations has a computational price; hence three-particle systems are used in this work as they represent the simplest, complex problem in many-body quantum physics and chemistry. Only by doing the calculations in full, is it possible to understand the validity of these standard quantum chemical assumptions.

There are a number of current authors who have contributed high accuracy results of three-particle systems. Frolov uses a double exponential basis set in perimetric coordinates for his wavefunction with 28 variational parameters to produce high accuracy energy calculations for atomic and molecular systems, along with accurate relativistic and QED¹ corrections^[34–41]. Nakashima’s high accuracy calculations for atomic and molecular systems^[28–30], are produced using the free iterative complement interaction (ICI) method. This uses a logarithm basis set in Hylleraas coordinates to produce the most accurate energy results for helium to date at over 40 digits. This accuracy requires a high number of basis functions (10000+), a high CPU time (4+ days) and a high level of decimal accuracy carried throughout the calculation (160+ digits)^[28]. Drake is another key author who has produced high accuracy calculations of relativistic effects and QED corrections for two-electron atoms using a triple exponential basis set^[42–45] along with high accuracy calculations of the excited state of two-electron atoms^[46]. Drake has also conducted high accuracy calculations for the critical nuclear charge, Z_C , which is the minimum charge required for binding two electrons to a nucleus^[47].

This work aims to complement and advance this previous research by using a compact series solution method requiring only 1-3 Non Linear Variational parameters (NLPs) to solve the time independent, non-relativistic Schrödinger equation obtaining high accuracy energies and wavefunctions, which are used to study fundamental quantum chemical physics. Chapter 2 describes the methods used in this thesis and Chapter 3 details the computational implementations of these methods.

The quantum theory of nuclear motion has remained underdeveloped compared to advancements in electronic structure theory and as experimental accuracy improves, it is important to make sure that theory can remain competitive, and able to model systems as accurately as possible using the fewest assumptions. Chapter 4 presents results of the effect of nuclear motion in three-particle diatomic molecules where the Born-Oppenheimer approximation is not assumed. Instead the coupling of electron-nucleus motions is included which means the non-BO results include the Zero Point Energy (ZPE) of the vibrational motion and kinetic energy of the nucleus along with the anharmonicity of vibrational motion.

Chapter 5 presents high accuracy electron correlation data, even for low nuclear charge Z systems. Particularly desirable is high accuracy data in the low density regime for both rigorous testing of long-range behaviour in newly developed methods and for the development of correlation functionals for use in density functional theory (DFT). A new Hartree-Fock implementation is presented which uses a Laguerre

¹Quantum electrodynamics (QED)

polynomial basis; and these results are compared to the fully-correlated methodology in order to study Coulomb holes for the lithium cation, helium atom, hydride anion. Additionally, the system with the critical nuclear charge for binding two electrons is considered to model the long-range, low density behaviour of the intracule density. The Coulomb holes for these systems were identified and characterised.

Experimentally, atoms and molecules in high Rydberg states, in which one electron is excited into an orbital far from the nucleus, hold importance in studies of chemical dynamics^[48]; whilst recent advances in the control of Rydberg-state atoms have led to possible new applications in quantum computing^[49]. Chapter 6 presents an implementation of a modified Pekeris method^[50] for excited states and is extended to include nuclear motion and variational parameters. This methodology is used to investigate the critical nuclear charge required to bind two-electron singlet and triplet excited states.

It is known that helium is an atom and the hydrogen molecular ion (H_2^+) is a molecule, but what of the positronium negative ion (Ps^-)? This system cannot be assigned a standard label of atom or molecule, on account of all three masses being equal. Chapter 7 investigates the transition from an atom-like system to a molecule-like system to better understand what it means to describe a system as an atom or a molecule.

CHAPTER 2



Theory

Chapter summary

The aim of this chapter is to provide a succinct overview of the series solution method used to approximately solve the three-particle Schrödinger equation; along with descriptions of the underlying mathematics used. Methods for checking the quality of the output wavefunctions are discussed followed by various particle densities that are calculated in this work. Finally an overview of the Born-Oppenheimer approximation and Hartree-Fock theory is given.

2.1 Atomic Units

Throughout this work atomic units (a.u.) are used, where key physical constants are set to unity i.e.

$$m_e = e = (4\pi\epsilon_0)^{-1} = \hbar = a_0 = 1$$

Their SI values are given in Table 2.1 and are taken from the latest CODATA values^[51].

Table 2.1 Various fundamental constants and their standard representations^[52].
Values in SI units.

Physical constant	Value in SI
Electron mass, m_e	$9.109\,383\,56(11) \times 10^{-31}$ kg
Electron charge, e	$1.602\,176\,620\,8(98) \times 10^{-19}$ C
Coulomb's constant, k_e	$1/4\pi\epsilon_0 = 8.987\,551\,787 \times 10^9$ Nm ² C ⁻² s
Reduced Planck constant, \hbar	$1.054\,571\,726(47) \times 10^{-32}$ Js
Bohr radius, a_0	$4\pi\epsilon_0\hbar^2m_e^{-1}e^{-2} = 0.529\,177\,210\,92(17) \times 10^{-10}$ m
Hartree, E_h or a.u.	$m_e e^4 (4\pi\epsilon_0)^{-2} \hbar^{-2} = 4.359\,744\,34(19) \times 10^{-18}$ J = 2625.49(96) kJmol ⁻¹

Alongside atomic units, (a.u.) muon atomic units are used for systems where the muon is the lightest mass and are scaled against the mass of the muon instead of the electron, i.e. $m_\mu = 206.768\,282\,6$ a.u. which is set to unity, $m_\mu = 1$, in muon atomic units, (m.a.u.).

2.2 Non-Relativistic, Time Independent Schrödinger Equation for Three-Particle Systems

The full, non-relativistic, time-independent Schrödinger equation in its simplest form can be written as

$$\hat{H}\Psi = E\Psi, \quad (2.1)$$

In this thesis, \hat{H} represents the Hamiltonian for a general three-particle system with particles of mass, m_i and charge, Z_i , $i = 1, 2, 3$. Using inter-particle coordinates, r_i where r_1 and r_2 are the distances between the oppositely charged particles and r_3 is the separation between the like-charged particles, the Hamiltonian takes the form (in atomic units)

$$\hat{H} = \underbrace{-\frac{1}{2m_1}\nabla_1^2 - \frac{1}{2m_2}\nabla_2^2 - \frac{1}{2m_3}\nabla_3^2}_{\text{Kinetic energy}} + \underbrace{\frac{Z_1Z_3}{r_1} + \frac{Z_2Z_3}{r_2} + \frac{Z_1Z_2}{r_3}}_{\text{Potential energy}}. \quad (2.2)$$

In this work the centre of mass motion is separated off to give the interaction Hamiltonian with particle 3, taken as the origin^[53]; although any of the particles can be chosen^[54]

$$\hat{H} = -\frac{1}{2\mu_{13}}\nabla_1^2 - \frac{1}{2\mu_{23}}\nabla_2^2 - \frac{1}{2m_3}\nabla_1 \cdot \nabla_2 + \frac{Z_1Z_3}{r_1} + \frac{Z_2Z_3}{r_2} + \frac{Z_1Z_2}{r_3}, \quad (2.3)$$

where

$$\mu_{13} = \frac{m_1m_3}{m_1 + m_3}, \quad \mu_{23} = \frac{m_2m_3}{m_2 + m_3}. \quad (2.4)$$

2.2.1 Inter-particle and Perimetric Coordinates

A three-particle system can be intuitively described using the inter-particle coordinate system, r_i , $i = 1, 2, 3$, where for atomic systems it is usual to label r_3 as r_{12} , Figure 2.1 (a). For molecules r_3 is used which represents the internuclear distance. The inter-particle coordinates have the undesirable property of being bound by the triangular condition^[13]. Two inter-particle coordinates have independent domains $r_i : 0 \rightarrow \infty$, $i = 1, 2$ whilst the third one is dependent on the other two $r_{12} : |r_2 - r_1| \rightarrow r_2 + r_1$. Coolidge and James introduced the perimetric coordinate system^[13] denoted here as z_i which are linear combinations of the inter-particle coordinates and represented schematically in Figure 2.1 (b).

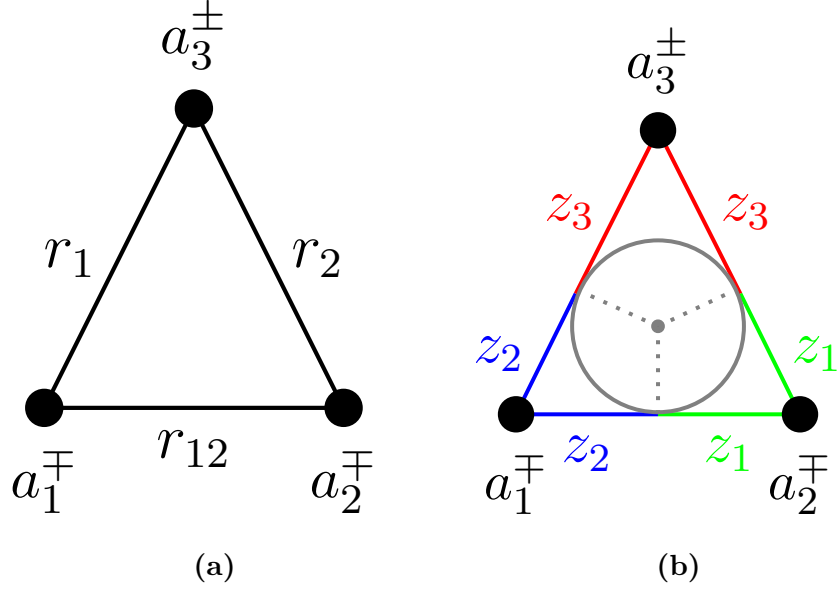


Figure 2.1 Coordinate systems used for a unit charge three-particle system (a) inter-particle coordinates r_i , (b) Perimetric coordinates, z_i . The in-circle sub-divides the sides of the triangle into perimetric coordinates.

Perimetric coordinates have the following form

$$z_i = r_j + r_k - r_i. \quad (2.5)$$

Their relation to inter-particle distances, r_i is given by

$$r_i = \frac{z_j + z_k}{2}, \quad (2.6)$$

where $\{i, j, k\}$ are cyclic permutations of 1,2,3. The benefit of using perimetric coordinates is that they are independent meaning no inter-dependency of the coordinates offering a great computational advantage, and they are valid in the interval $[0, \infty)$.

2.2.2 Wavefunction form

The wavefunction used in this work is based on the original work of Pekeris^[12] for two-electron atoms and subsequent modifications^[14,16,21,19,20]. It is fully correlated meaning it includes the inter-electronic distance r_{12} term in the basis set. The choice of basis is a triple orthogonal set of Laguerre polynomials. The Laguerre polynomial, $L_n(x)$ has a variety of definitions with the most common being the Rodrigues representation^[55]

$$L_n(x) = \frac{1}{n!} \left(\frac{d}{dx} - 1 \right)^n x^n. \quad (2.7)$$

Plotting the first few Laguerre polynomials in Figure 2.2 highlights how they each have very unique shapes.

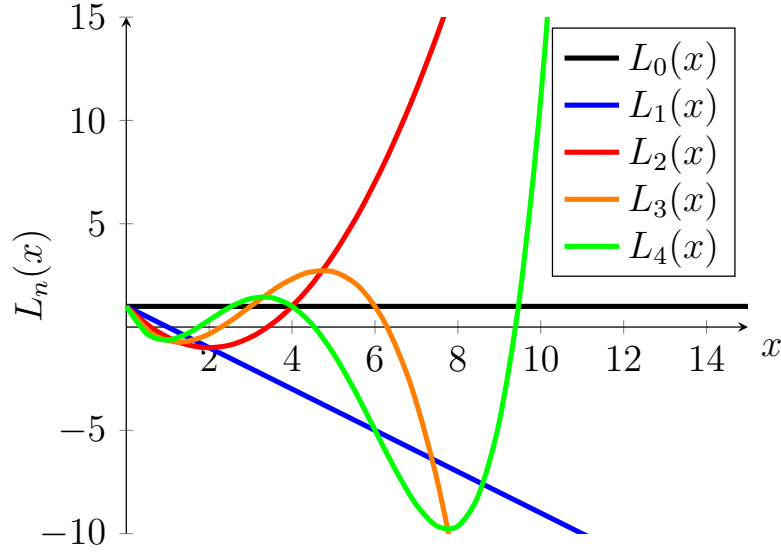


Figure 2.2 Plot of the first 5 Laguerre polynomial functions $L_n(x)$.

The wavefunction has the form given in equation (2.8)

$$\Psi(z_1, z_2, z_3) = e^{-\frac{1}{2}(\alpha z_1 + \beta z_2 + \gamma z_3)} \times \sum_{l,m,n=0}^{\infty} A(l, m, n) L_l(\alpha z_1) L_m(\beta z_2) L_n(\gamma z_3), \quad (2.8)$$

where α, β and γ represent non-linear variational parameters (NLPs) varied in order to minimise the energy and z_i represent perimetric coordinates. When combined together with an optimised scaling coefficient $A(l, m, n)$, the Laguerre polynomials will “mold” to better fit the true wavefunction. Ideally we would like to use every possible combination of $\{l, m, n\}$ but this is unattainable, thus it is truncated to order k . The more terms that are included in the wavefunction, the more accurate the approximation to the true wavefunction albeit at a computational price.

2.2.3 Fermionic Exchange

The wavefunction form given in equation (2.8) does not consider the sign of the overall wavefunction, so is only a valid form when all three-particles are different or bosons. When two identical fermions undergo exchange the sign of the total wavefunction changes which can be represented using a parity operator, \hat{P} . Applying this operator to the wavefunction twice (\hat{P}^2) should result in the wavefunction being unchanged as the particles are back where they started, i.e.

$$\hat{P}^2 \Psi(1, 2) \rightarrow \Psi(1, 2). \quad (2.9)$$

Thus $\hat{P}^2 = 1$ and the eigenvalues of \hat{P} are $p = \pm 1$, so to include the effect of parity, the perimetric coordinates, z_1 and z_2 representing the identical particle are interchanged

$$\Psi(z_1, z_2, z_3) \rightarrow \Psi(z_1, z_2, z_3) \pm \Psi(z_2, z_1, z_3), \quad (2.10)$$

resulting in the following new form of the wavefunction

$$\begin{aligned} \Psi(z_1, z_2, z_3) = e^{-\frac{1}{2}(\alpha z_1 + \beta z_2 + \gamma z_3)} \times \sum_{l,m,n=0}^{\infty} A(l, m, n) [L_l(\alpha z_1) L_m(\beta z_2) L_n(\gamma z_3) \\ \pm L_l(\alpha z_2) L_m(\beta z_1) L_n(\gamma z_3)]. \end{aligned} \quad (2.11)$$

Singlet states have an anti-symmetric spin function which means their spatial wavefunction is symmetric to exchange ($p = +1$) whilst the spatial wavefunction for triplet states is anti-symmetric to exchange ($p = -1$) as they have a symmetric spin function. Equation (2.11) is a correct wavefunction form for systems with two identical fermions where $+$ is valid for singlet states and $-$ for triplet states. The Laguerre orthogonality condition seen in equation (2.12)

$$\int_0^{\infty} e^{-x} L_m(x) L_n(x) dx = \delta_{m,n} = \begin{cases} 1 & \text{if } m = n \\ 0 & \text{if } m \neq n \end{cases} \quad (2.12)$$

is exploited in the formation of the secular determinant. Therefore for systems with two identical fermions the condition $\beta = \alpha$ must be enforced as terms arise which do not satisfy this condition. For systems where all three particles are unique or bosons all three NLPs can be varied independently. Modifications to the theory are possible to break this constraint which are discussed in Chapter 6, but within the ground state methodology, α and β must always be set equal to exploit the useful orthogonality, equation(2.12).

2.2.4 Non-Linear Variational Parameters: Wavefunction forms

The general wavefunction form is given in equation (2.8) and with symmetry considerations for singlet and triplet states in equation (2.11). There are various wavefunction forms used in this work that differ by the number of Non-Linear variational Param-

ters (NLPs) and how these effect the exponent and the basis. The correct long-range asymptotic behaviour for an atomic wavefunction is given by^[7,53]

$$e^{-\frac{k}{2}(r_1+r_2)}. \quad (2.13)$$

Hylleraas showed^[7] that the asymptotic behaviour of the solution of the Schrödinger equation for the helium atom requires that in the exact solution, $k = 2\sqrt{-E}$, where E is the electronic energy. Pekeris set $\epsilon = +\sqrt{-E}$ to ensure the wavefunction has the proper asymptotic form^[12], i.e.

$$(u + v + w) = 2\epsilon(r_1 + r_2), \quad (2.14)$$

where u, v and w represent scaled perimetric coordinates, $u = \alpha z_1, v = \beta z_2, w = \gamma z_3$. Pekeris' exponent form has no NLPs to assist with convergence, which was improved later by Cox *et al.* where a single NLP was added, K to aid in energy convergence^[16]

$$e^{-\sqrt{\frac{E}{K}}(r_1+r_2)}, \quad (2.15)$$

and as in the work of Pekeris, the conditions, $\alpha = \beta = \epsilon$ and $\gamma = \alpha + \beta = 2\epsilon$ scale the perimetric coordinates in the wavefunction, so when $K = 1$, the exponential term reduces to the correct asymptotic form given in equation (2.14)

$$(\alpha z_1 + \beta z_2 + \gamma z_3) = 2\epsilon(r_1 + r_2) = \left(\frac{2}{\sqrt{K}}\right) \sqrt{-E}(r_1 + r_2). \quad (2.16)$$

Both these wavefunctions have been implemented in this work, however modified forms are used following the work of Gálvez *et al.*^[19,20] where they removed the energy term, ϵ from the basis. The three wavefunctions used in the ground state work are referred to as $\Psi(\alpha), \Psi(\alpha, \gamma), \Psi(\alpha, \beta, \gamma)$ named for the number of NLPs which are allowed to vary independently. $\Psi(\alpha)$ has a single NLP, α with the parameter constraint, $\beta = \alpha, \gamma = \alpha + \beta = 2\alpha$

$$\Psi(\alpha) : e^{-\frac{1}{2}\alpha(z_1+z_2+2z_3)} \equiv e^{-\alpha(r_1+r_2)}. \quad (2.17)$$

Molecular-like systems suffer from poor convergence with the forms discussed so far due to no explicit dependence on the $r_{12} \equiv r_3$ coordinate, which represents a significant interaction for molecule-like systems. To account for this interaction, the constraint, $\gamma = \alpha + \beta$ is removed allowing for it to be used as a Non-Linear variational Parameter (NLP), giving $\Psi(\alpha, \gamma)$ which has two NLPs, α and γ with the parameter constraint, $\beta = \alpha$

$$\Psi(\alpha, \gamma) : e^{-\frac{1}{2}(\alpha(z_1+z_2)+\gamma z_3)} \equiv e^{-\frac{1}{2}\gamma(r_1+r_2)+(\frac{1}{2}\gamma-\alpha)r_{12}}. \quad (2.18)$$

This form is more suitable for molecular systems with two identical fermions; however is also valid for atomic systems. When $\gamma = 2\alpha$, this is equivalent to the form in equation (2.17).

When all three particles are distinct or bosons (e.g. HD^+) all three parameters can vary individually and thus one can use the following form with no parameter constraints

$$\Psi(\alpha, \beta, \gamma) : e^{-\frac{1}{2}(\alpha z_1 + \beta z_2 + \gamma z_3)} \equiv e^{-\frac{1}{2}r_1(\beta + \gamma - \alpha) - \frac{1}{2}r_2(\alpha + \gamma - \beta) - \frac{1}{2}r_{12}(\alpha + \beta - \gamma)}. \quad (2.19)$$

When $\alpha = \beta$ this form resembles equation (2.18) and when $\alpha = \beta, \gamma = 2\alpha$ the form in equation (2.17)

2.2.5 Series Solution Method

One of the other key reasons Laguerre polynomials are used is that they have a set of recursion relations that relate Laguerre polynomials, $L_n(x)$, to their first and second order derivatives, $L'_n(x), L''_n(x)$ ^[12]

$$\begin{aligned} xL_n(x) &= -(n+1)L_{n+1}(x) + (2n+1)L_n(x) - nL_{n-1}(x) \\ xL'_n(x) &= nL_n(x) - nL_{n-1}(x) \\ xL''_n(x) &= (x-1)L'_n(x) - nL_n(x) \end{aligned} \quad (2.20)$$

Pekeris^[12] developed an elegant series solution method which uses these relations to solve the heliogenic two-electron Schrödinger equation. Pekeris' original method was only capable of modelling atoms with a clamped nucleus, but modifications by other authors have allowed for the explicit inclusion of nuclear motion along with multiple variational parameters^[14,16,19,21] which also allows for the calculation of any three-particle systems including molecules.

Firstly the wavefunction, equation (2.8) is substituted into the Schrödinger equation. The Laguerre recurrence relations are then substituted into this resultant form which eliminates all first and second order partial derivatives of the Laguerre polynomials. Applying the Laguerre orthogonality condition, equation (2.12), results in a 57-term recursion relation (33-term for the fixed nucleus case^[12]) in terms of the coefficients

$$\sum_{a,b,c=-2}^{+2} C_{a,b,c}(l, m, n) A(l+a, m+b, n+c) = 0 \quad (2.21)$$

Equation (2.21) represents the solution to the three-particle Schrödinger equation and the sum ranges from -2 to 2 due to the second order partial derivatives contained in the kinetic energy operator. It is used to form a secular determinant to order k .

An integer is assigned to each triple index $\{l, m, n\}$, $i (\equiv l, m, n)$ or $k (\equiv l + a, m + b, n + c)$ using one of three numbering schemes dependent on the symmetry of the system, with $\omega = l + m + n$ being the maximum order. The ASYM case assigns integers to all combinations of $\{l, m, n\}$, whereas the SYM and ANTISYM cases account for wavefunction symmetry.

Asymmetric (No symmetry):

$$k_{\text{ASYM}}(l, m, n) = \binom{l+m+n-2}{3} + \binom{l+m+1}{2} + \binom{l}{1} + 1 \quad (2.22)$$

Symmetric (Singlet):

$$k_{\text{SYM}}(l, m, n) = \frac{\omega(\omega+2)(2\omega+5)}{24} + \frac{1-(-1)^\omega}{16} + \frac{l+m}{2} + \frac{(l+m)^2}{4} + \frac{1-(-1)^{l+m}}{8} + l + 1 \quad (2.23)$$

Antisymmetric (Triplet):

$$k_{\text{ANTISYM}}(l, m, n) = \frac{\omega(\omega+2)(2\omega-1)}{24} - \frac{1-(-1)^\omega}{16} + l(m+n) + m \quad (2.24)$$

Examples of these numbering schemes are shown in Table 2.2 up to $k = 10$.

Table 2.2 ASYM, SYM and ANTISYM numbering schemes to assign an integer to each triple index, where $(\omega = l + m + n)$.

Asymmetric					Symmetric					Antisymmetric				
l	m	n	ω	k_{ASYM}	l	m	n	ω	k_{SYM}	l	m	n	ω	k_{ANTISYM}
0	0	0	0	1	0	0	0	0	1	0	1	0	1	1
0	0	1	1	2	0	0	1	1	2	0	1	1	2	2
0	1	0	1	3	0	1	0	1	3	0	2	0	2	3
1	0	0	1	4	0	0	2	2	4	0	1	2	3	4
0	0	2	2	5	0	1	1	2	5	0	2	1	3	5
0	1	1	2	6	0	2	0	2	6	0	3	0	3	6
1	0	1	2	7	1	1	0	2	7	1	2	0	3	7
0	2	0	2	8	0	0	3	3	8	0	1	3	4	8
1	1	0	2	9	0	1	2	3	9	0	2	2	4	9
2	0	0	2	10	0	2	1	3	10	0	3	1	4	10

The case for a matrix order of $\omega = 2$ has been highlighted in Table 2.2 for each numbering scheme along with the corresponding matrix size, k . The symmetric and antisymmetric cases are able to achieve the same matrix order with much smaller matrix sizes and $k_{\text{ASYM}} = k_{\text{SYM}} + k_{\text{ANTISYM}}$. Each matrix element is made up of a sum of coefficients, C_{ik} , where i and k denote the row and column of the matrix. These are given by substituting $i(\equiv l, m, n)$ and $k(\equiv l + a, m + b, n + c)$ into the recursion relation. The collapsing of the triple index results in

$$\sum_k C_{ik} B_k = 0, \quad (2.25)$$

where B_k is the eigenvector. The problem is then solved as a generalised eigenvalue problem

$$\sum_k (H_{ik} - E S_{ik}) B_k = 0 \quad (2.26)$$

The Laguerre orthogonality condition in equation (2.12) sets a large number of the matrix elements to zero and results in sparse representations for the matrices \mathbf{H} and \mathbf{S} . More details are provided on the implementation of this theory in Chapter 3.

2.3 Eigenvalue Problem

An operator acting upon a function produces another function. A unique case occurs when the function generated is proportional to the original one

$$\hat{A}\Psi \propto \Psi, \quad (2.27)$$

and introduction of a proportionality constant, C , converts this into an equality

$$\hat{A}\Psi = C\Psi. \quad (2.28)$$

This represents the form for a standard eigenvalue problem where the constant C represents the eigenvalues and Ψ the eigenvectors. In quantum mechanics, for every dynamical variable A represented by an operator \hat{A} there exists a corresponding eigenvalue equation

$$\hat{A}\Psi_n = a_n\Psi_n. \quad (2.29)$$

The eigenvalues a_n represent the possible measured values of the variable A . As this problem is quantum mechanical, a_n will only have a sub-set of allowed values whereas classically it would be continuous. Since all results from measurements are real numbers, this imposes that the operators representing physical observables are Hermitian as eigenvalues of a Hermitian operator are real.

If the wavefunction (eigenvector) is not necessarily normalised, then one can use the generalised eigenvalue problem where S is the overlap and defines the relationship between the set of basis vectors of a quantum mechanical system

$$\hat{A}\Psi_n = a_n S\Psi_n. \quad (2.30)$$

The generalised eigenvalue problem is the type used exclusively in this work to approximately solve the three-particle Schrödinger equation.

2.4 Expectation Values: Variational Theorem

Expectation values make up the basis of all comparisons between the predictions of quantum mechanics and experiment. They represent the average value of the quantity under consideration. For an un-normalised wavefunction, Ψ , the expectation value corresponding to an observable x ^[56] is

$$\bar{x} \equiv \langle \hat{x} \rangle = \frac{\langle \Psi | \hat{x} | \Psi \rangle}{\langle \Psi | \Psi \rangle}. \quad (2.31)$$

Equation (2.31) can be interpreted as the average value of x that one would expect to obtain from an infinite number of experimental measurements with no errors.

One of the primary problems in quantum mechanics is to determine eigenvalues and eigenvectors of a given Hamiltonian

$$\hat{H}\Psi = E\Psi \quad (2.32)$$

When the system consists of a large number of particles meaning analytical means are not possible, the variational principle, which is based on the expectation value of H , offers a very powerful tool to solve the problem and provide energies. Consider a Hamiltonian operator, \hat{H} which contains the complete set of orthonormal eigenstates, $\phi_0, \phi_1, \dots, \phi_n$ and corresponding eigenvalues, E_0, E_1, \dots, E_n where E_0 represents the ground state energy. Any state Ψ can be expanded in terms of these eigenstates

$$\Psi = \sum_{n=0}^{\infty} c_n \phi_n, \quad (2.33)$$

where c_n represent coefficients. Using the definition of the expectation value in equation (2.31), one can calculate the expectation value of the energy as

$$\frac{\langle \Psi | \hat{H} | \Psi \rangle}{\langle \Psi | \Psi \rangle} \geq E_0. \quad (2.34)$$

Equation (2.34) is the definition of the variational theorem^[56]. It states that the expectation value of the energy is an upper bound to the true ground state energy E_0 . This form is the foundation of approximate calculations of ground state energies. One can choose a trial wavefunction that has a dependency on one or more variational parameters, $(\alpha_1, \alpha_2, \dots, \alpha_s)$

$$\Psi_{\text{trial}} = \Psi(\alpha_1, \alpha_2, \dots, \alpha_s). \quad (2.35)$$

Values are substituted for the variational parameters and the energy calculated using equation (2.34). The variational parameters are then varied in order to minimise the energy to a target accuracy^[56]

$$\frac{\partial E_0(\alpha_1, \alpha_2, \dots, \alpha_s)}{\partial \alpha_i} = 0, \quad i = 1, \dots, s. \quad (2.36)$$

The resulting minimum is the best estimate to the ground state energy of the system. The key advantage of the variational principle is that a complicated differential equation such as equation (2.32) has been replaced by a minimisation problem which is well handled by a computer. The main disadvantage is that there is not an exact way of finding the trial functions that are to be used; and requires some intuition about the problem.

2.5 Virial Condition

The virial theorem is important in statistical mechanics and is prevalent throughout the physical sciences from astrophysics to quantum mechanics. The virial condition used in quantum mechanics is the same as the one used in classical mechanics albeit the kinetic and potential energies are replaced by their expectation value equivalents. The virial condition (η) for a Coulomb potential^[54] states that the potential energy and kinetic energy will have the following relationship:

$$\eta = \frac{\langle \hat{V} \rangle}{\langle \hat{T} \rangle} + 2 = 0 \quad (2.37)$$

This acts as a quality control test for the wavefunctions in this work as the expected values of the kinetic energy and potential energy calculated using the wavefunctions must satisfy this relationship.

2.6 Cusp Condition

When two particles coalesce the potential energy becomes singular, however the Hamiltonian must remain self adjoint and bounded^[57]. In order for this to occur the kinetic energy must diverge to compensate for the singularity caused by the potential energy. Kato rigorously proved a fundamental relationship between the kinetic and potential energy at the coalescence point under the heavy-nucleus approximation, referred to as the cusp condition^[57]

$$\left(\frac{\partial \hat{\Phi}}{\partial r_{12}} \right)_{r_{12}=0} = \gamma(\Phi)_{r_{12}=0}, \quad (2.38)$$

where r_{12} represents the separation distance between the coalescing particles; and $\hat{\Phi}$ is Φ averaged over a sphere at the singularity^[57,58]. $\gamma = 1/2$ for the electron-electron singularity and $-Z$ for a nucleus-electron singularity where Z is the nuclear charge. More generally the exact cusp value is^[57]

$$\nu_{ij} = Z_i Z_j \frac{m_i m_j}{m_i + m_j}, \quad (2.39)$$

where Z_i, Z_j are the particle charges and m_i, m_j the particle masses. Bingel removed the spherical average enforced in Kato's derivation^[59] whilst Pack and Byers Brown introduced an expansion method based on Bingel's form for calculating the cusp, assuming the wavefunction does not vanish at coalescence^[58]

$$\Phi = (\Phi)_{r_{12}=0}[1 + Z_1 Z_2 \mu_{12} r_{12} + \vec{r}_{12} \cdot \vec{u}_{12} + O(r_{12}^2)], \quad (2.40)$$

where μ_{12} is the reduced mass of the coalescing particle pair. The cusp condition, equation (2.39) is exact, so the wavefunctions calculated in this work have a reference value which they must agree with; thus the difference between the exact cusp and the calculated cusp is indicative of the quality of the wavefunction. The cusp represents the gradient of the wavefunction at coalescence and can be calculated using

$$\nu_{ij} = \left. \frac{\partial \Psi}{\partial r_{ij}} \right|_{r_{ij}=0} \quad (2.41)$$

This is more conveniently expressed as^[60,41]

$$\nu_{ij} = \langle \hat{\nu}_{ij} \rangle = \frac{\langle \Psi | \delta(\mathbf{r}_{ij}) \frac{\partial}{\partial r_{ij}} | \Psi \rangle}{\langle \Psi | \delta(\mathbf{r}_{ij}) | \Psi \rangle}. \quad (2.42)$$

In Chapter 3 it is shown that equation (2.42) is more accurate than the expansion form given in equation (2.40), and hence equation (2.42) is the primary implementation used in this work with a description of the code given in Chapter 3.

2.7 Particle Densities

With one of the key motivations in this thesis being to probe atomic and molecular systems using high accuracy wavefunctions, particle densities are an intrinsic part of the work to follow. A particle density is a probability density measured between two particles, which can be compactly written as^[61]

$$D_{P,a_1}^{(1)}(\mathbf{R}) = \langle \Psi | \delta(\mathbf{x}_{a_1} - \mathbf{x}_P - \mathbf{R}) | \Psi \rangle. \quad (2.43)$$

This characterises the spatial distribution of particle a_1 with respect to some fixed point P . For states with angular momentum $L = 0$ and parity $p = +1$ the wavefunction, and thus the particle densities, are spherically symmetric. Therefore $D_{P,a_1}^{(1)}$ for the particle densities are also spherically symmetric, and their values depend on the length of \mathbf{R} . Due to this spherical symmetry the form given in equation (2.43) can be simplified and the following is introduced^[62,23]

$$\rho_{P,a_1}(r) = D_{P,a_1}^{(1)}(\mathbf{R}) \quad (2.44)$$

with $\mathbf{R} = (0, 0, r)$ and $r = |\mathbf{R}|$, $r \in \mathbb{R}_0^+$. The fixed point P in equation (2.43) and (2.44) can be chosen to be the position of a_3 , a_2 or the centre of mass, C shown in Figure 2.3.

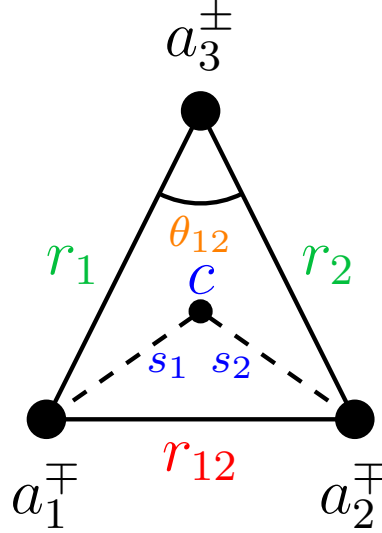


Figure 2.3 General three-particle diagram $\{a_1^\mp a_2^\mp a_3^\pm\}$ highlighting various particle densities calculated in this work including: **radial**, **intracule**, **centre of mass** and **angular** densities.

Their mathematical representations are as follows.

Radial density:

$$\rho(r) = \langle \Psi | \delta(\mathbf{r}_i - r) | \Psi \rangle, \quad i = 1, 2. \quad (2.45)$$

Intracule density:

$$h(r) = \langle \Psi | \delta(\mathbf{r}_{12} - r) | \Psi \rangle. \quad (2.46)$$

Centre of mass particle density:

$$\rho_{C, a_i}(r) = \langle \Psi | \delta(\mathbf{s}_i - r) | \Psi \rangle, \quad i = 1, 2. \quad (2.47)$$

Angular densities require a third point in order to define the angle^[61]

$$D_{P, a_1 a_2}^{(1,2)}(\mathbf{R}_1, \mathbf{R}_2) = \langle \Psi | \delta(\mathbf{x}_{a_1} - \mathbf{x}_P - \mathbf{R}_1) \delta(\mathbf{x}_{a_2} - \mathbf{x}_P - \mathbf{R}_2) | \Psi \rangle, \quad (2.48)$$

which is not portrayed by equation (2.43). The angular particle density is given by

$$\Gamma_{a_3, a_1 a_2}(\theta) = \langle \Psi | \delta(\theta_{12} - \theta) | \Psi \rangle, \quad \theta \in [0, \pi]. \quad (2.49)$$

The main component in calculating all these particle densities is the sifting theorem of the Dirac delta function^[63]:

$$\int_a^b f(x)\delta(x-c)dx = \begin{cases} f(c), & \text{if } a < x < b \\ 0, & \text{otherwise.} \end{cases} \quad (2.50)$$

When integrated, the product of any well-behaved function, $f(x)$ and the Dirac delta distribution yields the function evaluated where the Dirac delta function is singular^[64]. This is a useful property as it means the product $\Psi^*\Psi$ is evaluated at specific points using the Dirac delta function for any of the components of the wavefunction, producing a particle density along that component. Within the code written to calculate these particle densities, the variable under consideration, (r_i, θ_i) , etc... is varied using small incremental steps until the desired maximum is reached, which ensures a smooth, accurate particle density plot. A description of this code is provided in Chapter 3.

2.8 Born-Oppenheimer Approximation

One of the fundamental approximations used in many body quantum physics and chemistry is also one of the most intuitive. Very light particles move much faster than heavier ones, so to simplify the complicated many-particle Schrödinger equation Max Born and Robert Oppenheimer treated their motions independently^[65]. The approximation can be physically interpreted as the electrons following the nuclei's motion in an adiabatic manner; meaning they are “dragged” along by the nuclei without the need for a relaxation time. Mathematically this means separating the molecular wavefunction into its electronic and nuclear (vibrational, rotational, translational) components

$$\Psi_{\text{total}} = \Psi_{\text{electronic}} \times \Psi_{\text{nuclear}} \quad (2.51)$$

This approach means “freezing” the heavier nuclei in the molecule with a fixed internuclear separation, R_{ab} , then calculating the total electronic energy, U along with its associated wavefunction by solving the electronic Schrödinger equation in the field of these fixed nuclei. The electronic Hamiltonian for a system of N electrons and M nuclei has the form:

$$\hat{H}_{\text{el}} = \underbrace{-\frac{1}{2} \sum_{i=1}^N \nabla_i^2}_{\text{electron's kinetic energy}} - \underbrace{\sum_{i=1}^N \sum_{a=1}^M \frac{Z_a}{|r_{ia}|}}_{\text{electron-nuclear potential energy}} + \underbrace{\sum_{i=1}^N \sum_{j>i}^N \frac{1}{|r_{ij}|}}_{\text{electron-electron potential energy}} + \underbrace{\sum_{i=1}^M \sum_{b>a}^M \frac{Z_a Z_b}{|R_{ab}|}}_{\text{nucleus-nucleus potential energy}} \quad (2.52)$$

This process is then repeated for a series of values of R_{ab} to obtain the complete potential energy surface, $U(R_{ab})$. The total energy of the “unfrozen” molecule is then

$$E_{\text{total}} = E_{\text{electronic}} + E_{\text{vrt}} \quad (2.53)$$

Where $E_{\text{electronic}}$ is the electronic energy and E_{vrt} are the energy contributions from the vibrational, rotational and translational motions of the nuclei.

2.9 Hartree-Fock Theory

Hartree-Fock theory (HF) is one of the cornerstones of many-body quantum physics and chemistry and fundamental to much of electronic structure theory. It is the basis behind molecular orbital (MO) theory which attempts to describe the motion of each electron using a single-particle function which does not explicitly depend on the instantaneous motions of the other electrons in the system. HF theory often provides a firm starting point for more elaborate/advanced ideas such as many-body perturbation theory and single-reference configuration interactions^[66]. Modern chemistry has been built upon the orbital approximation so is testament to the power and appeal of HF theory.

When considering a many-electron problem, a many-electron basis is required which must satisfy two requirements:

1. Spin-orbitals need to be considered.
2. The chosen wavefunction must be antisymmetric with respect to electron exchange, so cannot be constructed as the product of one-electron spin-orbitals.

The original method of Hartree was to construct the wavefunction as the product of one-electron spatial wavefunctions^[67]

$$\Psi = \psi(1)\psi(2) \dots \psi(n) \quad (2.54)$$

This approach is flawed as it fails to account for electron spin, as when two electrons are exchanged the sign of the wavefunction must change (antisymmetry principle)

which does not occur in the Hartree product, and so modifications were introduced by Slater and Fock^[68]. The Slater wavefunction contains spin orbitals corresponding to spin up (α) and spin down (β) respectively for each spatial orbital. The spatial part of both spin states is the same, hence this discussion is valid only for the restricted HF approximation. For an unrestricted HF treatment the spatial part of the two spin functions is allowed to vary independently but is not discussed here. The Slater wavefunction is not a product of orbitals but is instead a determinant, known as the Slater determinant^[66]

$$\Psi = \frac{1}{\sqrt{(n)!}} \begin{vmatrix} \psi_1(1)\alpha(1) & \psi_1(1)\beta(1) & \cdots & \psi_n(1)\alpha(1) & \psi_n(1)\beta(1) \\ \psi_1(2)\alpha(2) & \psi_1(2)\beta(2) & \cdots & \psi_n(2)\alpha(2) & \psi_n(2)\beta(2) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \psi_1(n)\alpha(n) & \psi_1(n)\beta(n) & \cdots & \psi_n(n)\alpha(n) & \psi_n(n)\beta(n) \end{vmatrix} \quad (2.55)$$

Where $1, 2, \dots, n$ represent the coordinates of electron $1, 2, \dots, n$ respectively. Using a Slater determinant means that all electrons are indistinguishable as each electron is associated with every orbital. Moreover it automatically satisfies the antisymmetry principle with respect to the interchange of any two electrons, and the wavefunction disappears if two electrons have the same set of quantum numbers. A mean field approximation results in the electrons moving independently of all other electrons and a Coulomb repulsion is experienced due to the average positions of all electrons. Assuming that the electrons move independently is equivalent to assuming they can be described by a Slater determinant. Fock introduced the Fock operator^[68]

$$\hat{F} = \hat{H}^{\text{core}}(1) + \sum_{j=1}^n [2J_j(1) - K_j(1)], \quad (2.56)$$

Where the one-electron core Hamiltonian is

$$\hat{H}^{\text{core}}(1) = -\frac{1}{2}\nabla_1^2 - \sum_{a=1}^M \frac{Z_a}{r_{a1}} \quad (2.57)$$

J represents the Coulombic interaction of electron 1 with electron 2 in the orbital, Ψ_j

$$J_j(1)\psi_i(1) = \left\{ \int \psi_j^*(2) \frac{1}{r_{12}} \psi_j(2) d\tau \right\} \psi_i(1). \quad (2.58)$$

The exchange operator, K , which defines the quantum effect of electron exchange which is effectively a correction to J arising from same spin electrons avoiding one

another thus reducing the overall electrostatic repulsion

$$K_j(1)\psi_i(1) = \left\{ \int \psi_j^*(2) \frac{1}{r_{12}} \psi_i(2) d\tau \right\} \psi_j(1). \quad (2.59)$$

Each Fock operator calculates the energy of a single spatial orbital

$$\hat{F}\psi_n(1) = \epsilon_n\psi_n(1). \quad (2.60)$$

Written as a matrix equation

$$\hat{F} \begin{pmatrix} \psi_1(1) \\ \psi_2(1) \\ \psi_3(1) \\ \vdots \\ \psi_n(1) \end{pmatrix} = \begin{pmatrix} \epsilon_1 & 0 & 0 & \cdots & 0 \\ 0 & \epsilon_2 & 0 & \cdots & 0 \\ 0 & 0 & \epsilon_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \epsilon_n \end{pmatrix} \begin{pmatrix} \psi_1(1) \\ \psi_2(1) \\ \psi_3(1) \\ \vdots \\ \psi_n(1) \end{pmatrix} \quad (2.61)$$

This problem is solved using an initial guess for the wavefunction form. A new implementation of Hartree-Fock theory using a Laguerre polynomial wavefunction is discussed in Chapter 5 which describes the Hartree-Fock process in greater detail.

2.10 Summary and Conclusions

This chapter summarises some of the key theoretical and mathematical ideas underlying the work in this thesis. The next chapter will discuss the programming behind these methods.

CHAPTER 3



Computer Programming

3.1 Introduction

The work presented in this thesis required a substantial amount of programming which will now be discussed. It is not possible to describe all of the codes developed or their features due to their size and scope, but overviews of the main codes are given along with a brief walk-through of key ideas so readers can understand the program chain.

There are four sections which describe the key programs developed along with descriptions of how problems were solved and justifications for the methods used. First the C plus plus (C++)^[69] programs will be discussed, then Maple^[70] programs, followed by Python^[71] programs and finally Bash programs. C++ and Python are object oriented general purpose programming languages, Maple is a symbolic and numeric computing environment, as well as a multi-paradigm programming language and Bash is a Unix shell and command language.

The design philosophy used throughout the code development was to make use of open access resources from the programming and scientific community where available. Various libraries are used with examples including:

- Numerical Recipes^[72]
- LAPACK^[73]
- Blas^[74]
- Boost^[75]
- ExprTK^[76]

Relevant licenses for the libraries and packages used are found in Appendix B. Libraries such as LAPACK and Blas are the gold standard of linear algebra vector and matrix computations and the routines available within these libraries have been substantially tested by a wide community and thus possible bugs are minimal due to continued user feedback and maintenance.

3.2 Program map

The following programming map highlights the key coding branches.

C++, Maple and Python are used for calculations whereas Bash is used to build bridges between all the codes to make implementation, running and debugging easier but does no actual calculations. The main 3body code outlined in Section 3.4 is only compatible with Mac OS¹ and is not compatible with Linux or Windows. The Maple and Python codes are compatible with Mac OS, Linux and Windows, and the Bash scripts compatible with Mac OS and Linux.



Figure 3.1 Programming map highlighting the four main branches of development, C++, Python, Maple and Bash. G.s corresponds to ground state and E.s corresponds to excited state. * Python calls C routines to do the numerically intensive computations.

¹**Mac OS versions tested:** 10.6.8 Snow Leopard, 10.7.5 Mountain Lion, 10.9.5 Mavericks, 10.10.5 Yosemite, 10.11.6 El Capitan, 10.12.5 Sierra.

3.3 Hardware description

All the results presented in this thesis were calculated on one of two iMacs, with most program development done using a ThinkPad T420 laptop.

iMac (2015):

- i7 processor, 4GHz
- 32 GB DDR3 RAM
- 4 physical, 8 logical cores
- Mac OS 10.12.5 Sierra

iMac (2012):

- i5 processor, 2.7GHz
- 16 GB DDR3 RAM
- 2 physical, 4 logical cores
- Mac OS 10.11.6 El Capitan

ThinkPad T420 (2012):

- i5 processor, 2.5GHz
- 16 GB DDR3 RAM
- 2 physical, 4 logical cores
- Ubuntu 16.04 LTS

3.3.1 Build environment

Listed below are the program/software versions used in the latest implementation of the codes discussed.

C++

- Compiled for x86-64 architecture using clang++ Apple LLVM version 6.1.0
- Version: C++ 11
- Optimised using `-Os` compiler flag generating smallest, fastest build
- Parallelised using Apple's Grand Central Dispatch

Maple

- Version: 2016.0

Bash

- Version: 3.2.57(1)

Python

- Python version: 2.7.14 Anaconda custom
- IPython version: 5.1.0
- SciPy version: 1.0.0
- NumPy version: 1.13.3

3.4 C++

C++ is a logical choice of programming language and allows for programs to be modular with the use of templates, inheritance² and polymorphism³.

3.4.1 The 3Body code

The workhorse of the ground state problem is a code named 3Body. This code was originally written by a previous group member, Dr Andrew King^[77], and inherited by the current author where modifications and changes have been made over time, of which the key ones will be made clear. Throughout this thesis this code will be referred to as the “ground state code”, although technically when solving the ground state eigenvalue problem excited state eigenvalues are also produced^[21,50]. The ground state methodology is not suited to excited states, and a better method is discussed in Chapter 6, whose implementation is covered later in this chapter and will be referred to as the “excited state code”.

In Chapter 2 the theory uses α, β and γ as the Non Linear Parameters, whereas equivalent A, B and C are used, $A \equiv \alpha, B \equiv \beta, C \equiv \gamma$ in the code as they are more convenient when programming, so will be the notation used in this chapter.

The 3Body code calculates the energies, wavefunctions and various expectation values of three-particle systems. It is an executable which can be invoked from the command line, with the following syntax.

²The ability to define a class in terms of another class which allows for code to be easily re-used.

³This allows interaction with a C++ object without knowing what the object is. The object could have many forms defined by the user and descends from a generic base class.

C++: Single Point (SP) calculation of clamped nucleus helium

```
1 3Body Z1=-1 Z2=-1 Z3=2 m1=1 m2=1 m3=infinity A=1 C=2 -r
   ↪ RR.rr -s 2856 -SYM
```

This conducts a single point calculation on the clamped nucleus helium system using two NLPs A and C . The option `-r` tells the `3Body` code to use the Laguerre recursion relations which are held in the `RR.rr` file used to build the matrices. A feature implemented by the current author is the ability to pass a matrix to the `3Body` code using the option `-m` followed by the location of the matrix file, which is much slower due to read-time than the recursion relation option but allows for a wider array of generalised eigenvalue problems to be solved. This method was initially used to solve the excited state problem, however was superseded by the Python implementation discussed later in this chapter. The `-s` option specifies the number of terms in the wavefunction (size of matrix) and finally the symmetry of the system is defined using `-SYM`. There are three options to choose from, `ASYM`: No symmetry, `SYM`: Required for a singlet state or `ANTISYM`: Required for a triplet state.

In practise the `3Body` code is controlled by either a run file input (see Section 3.7.1), or the `AllMethods` bash script (see Section 3.7.1) which allows for a variety of more advanced options to be passed to the executable making calculations more efficient.

Precision

The `3Body` code was originally designed to be run at standard double precision (16 digit, binary64), however it was found that certain calculations warranted quadruple precision (32 digit, binary128) due to arithmetic precision losses. An example of this is given in Table 3.1 for the ground state triplet calculation of helium using two Non Linear variational Parameters (NLPs) (A, C) with fixed nucleus. The quadruple precision calculation was done as a single point (SP) calculation using the optimised A and C parameters from the double precision calculation for direct comparison.

Table 3.1 Energy comparison for the triplet ground state of helium with fixed nucleus calculated using double and quadruple precision. The quadruple precision calculation is a single point calculation performed with the A and C parameters obtained from the double precision optimisation calculation. Bold digits represent converged digits. Digits highlighted in red represent incorrect digits.

Matrix size	Double precision	Quadruple precision
946	-2.175 229 378 236 083	-2.175 229 378 236 084
1378	-2.175 229 378 236 710	-2.175 229 378 236 710
2600	-2.175 229 378 236 808	-2.175 229 378 236 785
Exact	-2.175 229 378 236 79 ^[29,39,78]	

As the energy calculations in this work are variational, the results obtained must not be more negative than the high accuracy energy calculations from authors such as Drake^[78], Frolov^[39] and Nakashima^[29] who all report and agree on a high number of converged digits. Different CODATA values for the helium nuclear mass will alter the results, so the nucleus was fixed to remove this dependency. The double precision calculation at a matrix size of 2600 in Table 3.1 is seen to be more negative than literature signalling that either the calculation is not variational, or there is precision loss. By calculating at quadruple precision, the accuracy is restored and reaffirms that the calculation is indeed variational.

Implementation of quadruple precision was done by the current author using the external library `mpfrc++`^[79] which contains the `mpreal` data type. `mpfrc++` is a C++ interface for the arbitrary precision data type of another external library `mpfr`. The `mpreal` data type is only limited by the machines memory hence is valid for arbitrary precision. The `mpfrc++` library used in this work is version 3.5.6 (Mercurial revision d8d1f40) under the General Public License (See Appendix B) and `mpfr` is version 3.1.2 under the Lesser General Public License (See Appendix B). Future modifications to the 3Body code would see the implementation of the `qd` library^[80] which is better optimised for quadruple precision calculations.

Running calculations at quadruple precision routinely is undesirable as they need to be run in software rather than hardware which *greatly* increases the computational time^[81]. When quadruple precision is required in this work, the system will first be optimised (OPT) at standard double precision and a quadruple precision single point (SP) calculation done using the results from the optimisation. This strikes a balance between accuracy and computational cost.

Building/loading the matrices

In the template main block of the C++ code the first step creates the 5 matrices required for the problem. The first three represent the kinetic, potential and overlap matrices \mathbf{T} , \mathbf{V} and \mathbf{S} with the last two labelled as \mathbf{hh} and \mathbf{ss} which are derived from the left and right hand side respectively of the generalised eigenvalue problem

$$\underbrace{(\hat{T} + \hat{V})}_{hh} \Psi = E \underbrace{S}_{ss} \Psi \quad (3.1)$$

The recursion relations generated in the Maple program outlined in Section 3.5.1 are read by 3Body and processed to create these 5 matrices. Markers are used, t , p , o , hh , ss in the recursion relation to correctly allocate terms to each matrix; with the markers themselves holding no mathematical significance and are equivalent to multiplying by 1.

1. t : Kinetic energy terms
2. p : Potential energy terms
3. o : Overlap terms
4. hh : LHS of generalised eigenvalue problem, equation (3.1)
5. ss : RHS of generalised eigenvalue problem, equation (3.1)

The 3Body program was designed to have one general matrix class called `Matrix`. This class contains all the mathematical expressions and will generate any of the 5 matrices upon request by the user. This highlights the advantage of using inheritance and polymorphism as this `Matrix` class will return any matrix and does not require coding separate classes for each of the 5 matrices. The `Matrix` class returns the matrices using the function `GetMatrix(MatrixPart Part)` where `Part` tells it to return the specified matrix, e.g. `GetMatrix(MatrixPart t)` to return the kinetic energy matrix.

Two methods can be used to generate the matrices, the first uses the class `Matrix_RR_JIT` to convert the recursion relations to numerics using a fork of the external library `mathpresso`, `doublepresso` (under the MIT license, see Appendix B) coupled with just in time (JIT) compilation of the mathematical expressions to assembled machine code (ASM) after the program has started as it has access to dynamic runtime information which is default for double precision calculations. The second method uses the class `Matrix_RR_NOJIT` which does

not use JIT compilation due to the mathpresso library not being compatible with any precision higher than double, so is the default for quadruple precision calculations and is much slower. Future implementations could see a modification to the mathpresso library to be compatible with the mpreal datatype in order to speed up arbitrary precision calculations.

C++: Using the Matrix class

```

1 Matrix<T> *TheMatrix = nullptr;
2 VariableSet<string> InitVars;
3 // Add the variables
4 InitVars.Add(Variable<string>("Z1", "1"));
5 InitVars.Add(Variable<string>("Z2", "1"));
6 InitVars.Add(Variable<string>("Z3", "2"));
7 // etc...
8 LMN::SymmType Symm = LMN::SymmType::Symmetric; //The
  ↪ wavefunction symmetry. This is needed to generate the
  ↪ matrix when using the recursions relations
9 MatrixRules* Rulesp = nullptr; //Rules for loading KE, PE,
  ↪ OV, hh, ss matrices from the Matrix class, i.e. what
  ↪ markers are used
10 KERHS=false;
11 vector<string> WithT{"t"};
12 vector<string> WithoutT(0);
13 vector<string> WithV{"p"};
14 vector<string> WithoutV(0);
15 vector<string> WithO{"ss"};
16 vector<string> WithoutO(0);
17 Rulesp = new
  ↪ MatrixRules(WithT, WithoutT, WithV, WithoutV, WithO, WithoutO,
  ↪ KERHS);
18 long RR_MatSize = 2856;
19 TheMatrix = new Matrix_RR_NOJIT<T>(FileName, RR_MatSize,
  ↪ Symm, Rules, InitVars); //Initialise the matrix to
  ↪ explicitly use the recursion relation of the Laguerre
  ↪ polynomials
20 //or
21 TheMatrix = new Matrix_MathJIT<T>(FileName, -1, Rules,
  ↪ InitVars); //Initialise the matrix to read a matrix from
  ↪ file

```

```

22
23 VariableSet<string> Vars;
24 //Variables can be added or changed using the following
25
26 Vars.Add(Variable<string>("A", "1"));
27 Vars.Add(Variable<string>("C", "2"));
28 TheMatrix->EnterVariables(VariableSet<string> Vars);
29 TheMatrix->GetMatrix(MatrixPart::HH)->Print(); //prints the
    ↪ Hamiltonian, HH matrix to terminal.

```

Evaluating matrix elements

Once the matrices have been built using the recursion relations ($-r$), or matrices read in from file ($-m$), all the matrix elements will be algebraic, and need to be evaluated with the user specified values (Z_i , m_i , NLPs). Firstly, the *Spirit* recursive parser from the *Blas* library is used to parse the text input into a numerical format. Once this is done the external library *doublepresso* is used to evaluate the resulting expression to give a numerical value as described above. This is a much faster approach than parsing the algebraic expressions as strings, however it is not coded to work with arbitrary precision whose implementation makes for possible future work.

Other possible future work would be to implement a different parser than *Spirit* as after profiling of the *3Body* code it was found $\sim 40\%$ of the *3Body* execution time was spent substituting the user variables into the matrix elements. An alternative math parser engine used later in this work, *ExprTK*^[76] could be a suitable alternative.

Solving the Generalised Eigenvalue Problem

There is a wealth of study into algorithms for solving eigenvalue problems. As the matrices in this work are symmetric, this symmetry can be exploited to aid in the solving of the generalised eigenvalue problem. There is a choice of two algorithms which have been implemented, QL decomposition implemented using numerical recipes^[82,83] and MRRR⁴ called from LAPACK function *dsyevr*^[73] using the *accelerate* framework bindings⁵. The MRRR is the default algorithm used in this work as it is faster than QL.

⁴Multiple Relatively Robust Representations.

⁵Mac OS only, future work would see the function called directly from LAPACK for use with Linux.

Calculation Options

There are three calculation types possible using the 3Body program:

1. **Optimisation (OPT):** Vary NLPs using algorithms in order to find the lowest minimum on the potential energy surface.
2. **Single Point (SP):** Calculate the energy with fixed values for all NLPs.
3. **Scan (SCAN):** Scan the potential energy surface by calculating the energy for small increments in all the NLPs. This is effectively many repeated single point calculations in order to visualise the PE surface.

The first of these options, optimise, will be discussed in more detail, as the other two are conceptually easier and less important. Non-linear minimisation/maximisation is a vast research topic covering physical sciences, informatics, economics and many other fields^[84–90]. Various optimisation algorithms have been implemented for the 3Body code in order to minimise the ground state energy by varying the NLPs. These algorithms are

1. **BOBYQA:** The Bound Optimisation BY Quadratic Approximation algorithm^[91] was found to be the most effective algorithm for atomic systems whilst performing poorly for molecular systems (see Tables 3.2 and 3.3). One of the benefits of this algorithm is its use of a trust region technique so does not require any gradient information. It instead needs finite bounds in which to locate the minimum.
2. **NROpt:** This is a variant of the Conjugate Gradient algorithm implemented from numerical recipes third edition^[72]. This was found to be very effective at locating the minimum on the PE surface.
3. **BFGS:** The Broydon-Fletcher-Goldfarb-Shanno algorithm^[92]. This is a quasi-Newton second derivative line search algorithm, and is sourced from the external `dlib` library^[93].
4. **L-BFGS-B:** This is a reduced memory variant of the BFGS algorithm^[92], once again sourced from the `dlib` library^[93]. This algorithm was designed for numerical problems dependent on many NLPs, so has not been required for the work thus far, but is a very powerful algorithm included for possible future work.

Table 3.2 compares these algorithms for the calculation of the ground state energy of helium with nucleus in motion using two NLPS (A, C).

Table 3.2 Ground state energy of He calculated using four optimisation algorithms (listed in order of accuracy, see text for their details) using a 444 size matrix and two NLPs (A, C).

Algorithm	A		C		Energy
BOBYQA	3.101	537	5.334	798	-2.903 304 557
NROpt	1.796	940	1.913	710	-2.903 304 530
LBFGS	1.414	116	2.828	232	-2.903 304 467
BFGS	1.414	116	2.828	232	-2.903 304 467

Table 3.3 compares these algorithms for the calculation of the ground state energy of the hydrogen molecular ion using two NLPS (A, C).

Table 3.3 Ground state energy of H_2^+ calculated using four optimisation algorithms (listed in order of accuracy, see text for their details) using a 444 size matrix and two NLPs (A, B).

Algorithm	A		C		Energy
NROpt	7.526	690	1.380	706	-0.597 123 838
LBFGS	5.378	790	1.555	764	-0.597 102 546
BFGS	3.645	904	1.736	274	-0.596 993 098
BOBYQA	2.933	747	1.871	691	-0.596 834 788

BOBYQA performs the best for atomic helium whereas NROpt performs best for the hydrogen molecular ion. To achieve better convergence for molecular systems, a combination of BOBYQA and NROpt is now discussed.

Fork Method

It was found that the NROpt algorithm was the best at locating the position of the minimum on the PE surface, but BOBYQA was the best at minimising down to the trough of the minimum^[77]. This led to a combination scheme of NROpt and BOBYQA called the Fork method. This is the optimisation scheme used in this work for molecular systems where there are multiple minima on the PE surface. Both calculations run in parallel making use of GNU Parallel^[77] and once both are complete the better of the two methods with regard to energy stability is chosen. A

final BOBYQA is then conducted starting from the best NLP values. Table 3.4 shows two fork calculations for He and H_2^+ , and whilst no accuracy is gained for helium it offers the lowest minimum for H_2^+ when compared to single optimisations using NROpt and BOBYQA.

Table 3.4 Ground state energies calculated using fork optimisation method for He and H_2^+ using a 444 size matrix and two NLPs (A, C).

System	A	C	Energy
He	3.101 533	5.334 790	-2.903 304 557
H_2^+	7.519 752	1.359 622	-0.597 123 856

The energy result for H_2^+ obtained from the fork optimisation in Table 3.4 is more accurate than the one obtained using NROpt or BOBYQA from Table 3.3. BOBYQA tends to get stuck in local minima on the molecular PE surface, whereas NROpt is more capable of locating the global minimum. Table 3.2 shows BOBYQA performs the best for atomic systems where there is a single well defined minimum on the PE surface. A combination of the two algorithms (NROpt, BOBYQA) for molecular systems allows for the global minimum to be found (NROpt) and then minimised down to the trough (BOBYQA) more effectively.

An example of how to use the optimise class is now given

C++: Using the Optimise class

```

1 ParameterSearcher<T> *Method; //Create pointer to base class
  ↳ Parameter searcher
2
3 long tol = -3; //Set tolerance to  $3 \times 10^{-3}$ 
4
5 VariableSet<string> InitVars;
6 InitVars.Add(Variable<string>("m1", "1"));
7 InitVars.Add(Variable<string>("m2", "1")); // etc...
8 OptVarSet<T> OptVars; // This object holds the variables
  ↳ which are to be optimised whilst keeping track of their
  ↳ initial and final values
9 OptVars.Add(OptVar<T>("A", 1)); // Set initial values for NLPs
10 OptVars.Add(OptVar<T>("C", 2));
11 Method = new Optimise<T>(InitVars, OptVars, TheDiagonaliser
  ↳ , tol); // Create class Optimise

```

```

12 Optimise<T>* OptMethod = dynamic_cast<Optimise<T>*>(Method);
13
14 OptMethod->SetOptimisationMethod(NROpt); // Specify the
   ↪ algorithm to be used
15 OptMethod->SetOptimisationMethod(dLib_bobyqa);
16
17 Method->Maximise(false); // Whether to minimise or maximise
   ↪ the objective function
18
19 EigenResults<T> Results = Method->Process(); // This
   ↪ function does all the required steps for the
   ↪ optimisation
20
21 std::cout << "The eigenvalue was " <<
22 Results.LowestEigen().EigenValue <<
23 " with Energy of " << Results.LowestEigen().Energy << "\n";
   ↪ //Print the results

```

3.4.1.1 3Body Code Validation

With the outline of methods and techniques used described in the previous section, it is imperative to show it works correctly. An extensive validation is not presented here as previous publications^[21,22,94,95,23] are proof enough; however for completeness, several test examples are now presented.

Atoms:

Table 3.5 Energies (a.u.) calculated using the 3Body code with the BOBYQA algorithm for the helium isoelectronic sequence, $Z = 1, \dots, 7$ with and without nuclear motion. A 2856 size matrix ($\omega = 30$) was used for all systems. The finite mass values, M , are taken from CODATA 2014^[52].

System	Energy (a.u.) $m_3 = \infty^{a,b}$	Energy (a.u.) $m_3 = M^{a,b}$
H ⁻	-0.527 751 016 54	-0.527 445 881 10
He	-2.903 724 377 03	-2.903 304 557 729
Li ⁺	-7.279 913 412 66	-7.279 321 519 8
Be ²⁺	-13.655 566 238 4	-13.654 709 268 24
B ³⁺	-22.030 971 580 2	-22.029 846 048
C ⁴⁺	-32.406 246 601 89	-32.404 733 488
N ⁵⁺	-44.781 445 148 7	-44.779 658 349 4

^a Bold digits in agreement with^[29].

^b All digits presented are converged.

Molecules:

Table 3.6 Energies calculated using the 3Body code for various electronic (atomic units, a.u.) and muonic molecular systems (muon atomic units, m.a.u.). Symmetric systems with two fermions where $m_1 = m_2$ use a 2856 size matrix and unsymmetric systems where $m_1 \neq m_2$ use a 5456 size matrix, both $\omega = 30$. The fork optimisation method was used for all optimisations. Particle mass values are taken from CODATA 2014^[52].

System	Energy (a.u.) ^{a,b}	System	Energy (m.a.u.) ^{a,c}
μ_2^+	-0.585 126 098 25	Mu ⁺	-0.262 005 070 23
H ₂ ⁺	-0.597 139 063 07	μH_2^+	-0.494 386 812 860 9
D ₂ ⁺	-0.598 788 784	μD_2^+	-0.531 111 130 611 07
T ₂ ⁺	-0.599 506	μT_2^+	-0.546 374 222 033 39
HD ⁺	-0.597 897 968 60	μHD^+	-0.512 711 790 563
HT ⁺	-0.598 176 134 6	μHT^+	-0.519 880 084 536 7
DT ⁺	-0.599 130 661	μDT^+	-0.538 594 970 881 1

^a All digits presented are converged.

^b Bold digits in agreement with Nakashima *et al.*^[30].

^c Bold digits in agreement with Frolov^[40].

Alongside this, a Python/C application⁶ of the ground state problem (currently only for atoms), implemented by the current author, agrees with the results calculated by the C++ code whilst being computationally faster. Figure 3.2 (a) shows a plot of CPU time vs. matrix size for two situations:

1. Ground state optimisation of helium with an infinite nuclear mass using one NLP, A .
2. Ground state optimisation of helium with an infinite nuclear mass using two NLPs, A, C .

Figure 3.2 (b) shows the same situations (1) and (2) but calculated using the Python implementation. All optimisations are conducted using the BOBYQA algorithm^[91] at double precision.

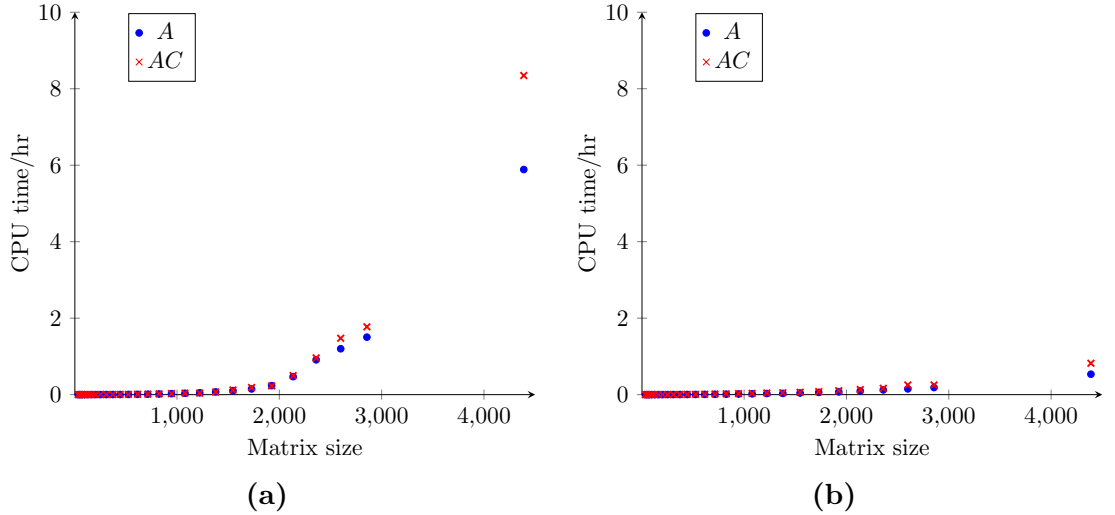


Figure 3.2 CPU time vs. matrix size for (a) C++ and (b) Python using 1 and 2 NLPs for the optimisation of clamped nucleus ground state helium using the BOBYQA algorithm.

3.4.2 Expectation Values

The Expectation code is designed to calculate expectation values of certain operators for which a recursion relation can be generated with examples being $\langle \hat{T} \rangle$, $\langle \hat{V} \rangle$, $\langle r_i^n \rangle$, where $n = -1, 1, 2, 3$, $i = 1, 2, 3$. It uses the same `Matrix` class as the 3Body code, and the only extra information required is the output eigenvector from the 3Body program. This will not be covered in detail as it is just an extension of

⁶Section 3.6 discusses the excited state code implementation, which the Python ground state code is built upon.

the 3Body program, but a simplified minimal working example of how it calculates expectation values is now given. It reads the eigenvector from a text file then creates the matrices using the recursion relations (RR) generated for the specific operator in Maple. It then calculates the product of the matrix and eigenvector and outputs the result.

```

C++: Calculating expectation values

1 long RR_MatSize = M.GetSize();
2 T ExpValue = 0;
3 T OvValue = 0;
4 for (long x=0; x<RR_MatSize; x++)
5 {
6     for (long y=0; y<RR_MatSize; y++)
7     {
8         ExpValue+=Vec[x]*ExpMatrix[x][y]*Vec[y];
9         OvValue+=Vec[x]*OvMatrix[x][y]*Vec[y];
10    }
11 }
12 cout << "Expectation Value=" << DtoStr(ExpValue) << "\n";
13 cout << "Overlap Value=" << DtoStr(OvValue) << "\n";
14 cout << "<Exp>/<Ov>=" << DtoStr(ExpValue/OvValue) << "\n";

```

3.4.3 Numerical integration

There are some problems where analytical means of solving definite integrals are not feasible or too computationally demanding. This problem can be circumvented using numerical integration techniques, where an approximate solution to a specified accuracy is sought for the following definite integral

$$\int_a^b f(x)dx \quad (3.2)$$

There are several motivations for using numerical integration:

- An anti-derivative of the function may be known but does not consist of elementary functions.
- The integrand, $f(x)$, may consist of thousands of terms.
- The analytical solution may create numerical instability^[96], examined in the following example.

Consider the integration^[96]

$$I_n = \int_0^1 x^n e^{x-1} dx. \quad (3.3)$$

This yields to integration by parts resulting in the recurrence

$$I_n = 1 - nI_{n-1}. \quad (3.4)$$

By examining the integral, $0 < I_{n+1} < I_n$ and $\lim_{n \rightarrow \infty} I_n = 0$. Programming this solution at double precision reveals that as n increases, violent oscillations develop caused by numerical instability; for example, when $n = 21$, $I_{21} = 0$ but when $n = 22$, $I_{22} = -1 \times 10^{15}$ and when $n = 23$, $I_{23} = 1 \times 10^6$. Even if a closed form is found for an integration, it is still constrained by numerical limitations inherent in computations.

In this work numerical integration techniques find their main use in the calculation of particle densities or difficult expectation values due to their complexity and size of the wavefunctions used. As an example

$$h(r) = \int_0^\infty \int_0^\infty \int_{|r_1-r_2|}^{r_1+r_2} \Psi(r_1, r_2, r_3) \delta(r_3 - r) \Psi(r_1, r_2, r_3) dr_3 dr_2 dr_1, \quad (3.5)$$

represents the form used to calculate intracule densities in Chapter 4. $\Psi(r_1, r_2, r_3)$ is the wavefunction in inter-particle coordinates which consists of a large number of terms. The sifting theorem of the Dirac delta function is used in order to plot the particle density which means re-calculating this integral for thousands of values of r to achieve a smooth, accurate particle density plot. Solving this in Maple analytically is possible with use of the `int` command, but is time consuming and sometimes produces incorrect results due to floating point rounding error. Solving these complicated integrals using `int` passes all control over to the Maple kernel and it is difficult to extract out what Maple is doing to solve the calculation. The numerical rounding issues with `int` were especially prevalent in Chapter 5 where it incorrectly calculated a third Coulomb hole, but upon using numerical integration techniques and correctly accounting for precision shows that it does not exist within the accuracy of the methods employed.

The CUBA library^[97] offers four algorithms for multi-dimensional numerical integration: `vegas`, `suave`, `divonne` and `cuhre`. `Vegas`, `suave` and `divonne` use sobol quasi-random sampling, `mersenne twister` pseudo-random sampling or `ranlux` pseudo-random sampling techniques which can be specified by the user. They are all Monte

Carlo techniques which means they are not a good choice for the accuracy required in this work. All of the algorithms in CUBA only work on an N-dimensional unit hypercube; which means the integration domains must first be modified for the problem under consideration. Mathematically, this means finding an approximate solution to the following definite integral^[98]

$$\mathbf{I}[\mathbf{f}] = \int_{a_1}^{b_1} \int_{a_2}^{b_2} \cdots \int_{a_n}^{b_n} \mathbf{f}(\mathbf{x}) d\mathbf{x}. \quad (3.6)$$

Where \mathbf{x} is an n -vector and \mathbf{f} is an integrand. The N-dimensional unit hypercube is represented as, $[a_1, b_1] \times [a_2, b_2] \dots \times [a_n, b_n]$.

Cuhre is a deterministic algorithm⁷ which uses quadrature rules over n -dimensions (Genz-Malik cubature rules^[98] of polynomial degree). It can only process Riemann integrals of the form

$$\int_0^1 f(\mathbf{x}) d^d x, \quad (3.7)$$

where $f(\mathbf{x})$ can be sampled at arbitrary points, $x \in [0, 1]^d$. The integral is approximated using the following quadrature formula

$$Q_n f = \sum_{i=1}^n w_i f(x_i). \quad (3.8)$$

Where x_i represent nodes and w_i the weights of each node. An orthogonal basis of functions, $\{b_1, \dots, b_m\}$ is selected by the program which can sufficiently approximate f , and it is imposed that each b_i be integrated exactly by Q_n

$$\sum_{k=1}^n w_k b_i(x_k) = \int_0^1 b_i(\mathbf{x}) d^d x. \quad (3.9)$$

Cuhre uses a globally adaptive subdivision estimation routine which has the following steps provided as pseudo code:

1. Integrate entire region: $I_{\text{tot}} \pm \text{Total error}$
2. While Total error > desired maximum error
3. Find region, r , with largest error
4. Bisect this region
5. Integrate each subregion of r separately

⁷An algorithm which, given a particular input, will always produce the same output.

6. Update I_{tot}

7. end while

It is best suited to a low number of dimensions (1-5) and integrands which are well approximated by polynomials; both of which fit this work perfectly and hence it is the chosen algorithm.

Maple is used to change the integration limits, build and simplify the integrand which is then saved to a file in a C++ recognisable format. This file is then read into a C++ code which was written to insert this input into the Cuhre algorithm and the integration solved. The Maple program is discussed in Section 3.5 and snippets of the C++ code are printed below with important options and details on implementation.

C++: Import the integrand from file

```

1 // This links the long double variety of CUBA which needs to
  ↳ be compiled from source. A quadruple precision version
  ↳ can also be specified using cubaq.h
2
3 #include "cubal.h"
4
5 long double r3val;
6 // The integrand is processed as a function. The input must
  ↳ be recognisable by C++ as no input parsing is done at
  ↳ this stage. Future work could implement a regex parser.
7
8 // This example uses a function or two variables -> double
  ↳ integration for intracule density.
9 double myfunc(long double x, long double y) {
10     return
11     #include "../path to file containing integrand"
12     ;
13 }
```

Now the various options are specified such as number of dimensions, accuracies and number of integral iterations.

C++: Minimal working example of using the Cuhre algorithm input file

```

1 // Input options
2 #define NDIM 2 // The number of dimensions of the integral
3 #define NCOMP 1 // The number of computations to be done
```

```

4 #define NVEC 1
5 #define EPSREL 1e-13 // The desired relative error
6 #define EPSABS 1e-12 // The desired absolute error
7 #define VERBOSE 0 // Controls the amount of background the
  ↳ library will print, 0 gives just the result, 1,2 and 3
  ↳ give more details about each calculation step and
  ↳ subregion results.
8 #define SEED 0
9 #define MINEVAL 0 // The minimum number of integration
  ↳ iterations the program will do
10 #define MAXEVAL 50000000 // The maximum number of
  ↳ integration iterations the program will do
11 #define STATEFILE NULL
12 #define SPIN NULL
13 #define KEY 0 // Selects the Cubature rule
14
15 using namespace std;
16
17 int main()
18 {
19     FILE * pFile;
20     int comp, fail, nregions;
21     long long int neval;
22
23     pFile = fopen ("output_file", "w");
24
25     int t0 = time(NULL); // Set up the start of the timer
26
27     cubareal integral[NCOMP], error[NCOMP], prob[NCOMP];
28     // Now set the loop structure to calculate the particle
  ↳ density
29     for (r3val = 0.00001; r3val <= 20; r3val += 0.01) //
  ↳ Change the number of points here
30     {
31         llCuhre(NDIM, NCOMP, Integrand, USERDATA, NVEC,
32                 EPSREL, EPSABS, VERBOSE | LAST,
33                 MINEVAL, MAXEVAL, KEY,
34                 STATEFILE, SPIN,

```

```

35     &nregions, &neval, &fail, integral, error, prob);
36
37     // Print results to terminal and file
38     for(comp = 0; comp < NCOMP; ++comp)
39     {
40         printf("%.10Le\t%.20f\n", r3val,
41             ↪ (double)integral[comp]);
42         fprintf(pFile, "%.10Le\t%.20f\n", r3val,
43             ↪ (double)integral[comp]);
44     }
45 }
46 fclose (pFile);
47
48 int t1 = time(NULL);
49 printf ("time = %d secs\n", t1 - t0); // Print time
50     ↪ elapsed to the terminal
51
52 return 0;
53 }

```

This process requires that the C++ program be compiled every time a new integration is needed due to the integrand not having access to dynamic runtime information when inserted into a compiled program. This compilation is fast (order of seconds) but the whole process has been automated using a Bash script, with the key command being the compilation.

Bash: Compilation

```

1 g++ main.cpp.o -o Name_of_executable
  ↪ -L/../../path_to_cuba_library -lcubal

```

The long double version of the CUBA library allows for a maximum accuracy of 16-19 decimal places. If more accuracy is required then the quadruple precision (32 digits, float128) variety can be used but will take substantially longer as the calculation needs to be run in software rather than on hardware.

The numerical integration scripts to calculate particle densities were found to be ≈ 200000 times faster (real time) than calculating them using Maple's `int` command; and allowed for more control over the calculation as well as knowing the accuracy of the end result. Maple has an inbuilt CUBA implementation; but C++ was used to

optimise the speed of the calculation along with having greater control of the inner workings of the library itself.

3.5 Maple

The Maple codes developed in this work are not as fast or specialised as the C++ or Python codes, but what Maple lacks in computational speed it makes up for in speed of code development and symbolic mathematical power, which are the motivations for its use. Alternatives can be used such as symbolic C++^[99] or Sympy^[100] which are both open source, but lack the same development history that Maple has had and thus do not have as many features.

3.5.1 Recursion Relations

A Maple program calculates the series solution recursion relations by applying the Laguerre recurrence relations in order to solve the Schrödinger equation along with various expectation values. Maple does the following:

1. Define the operators in inter-particle coordinates r_i .
2. Perform a coordinate transformation from inter-particle to perimetric coordinates (z_1, z_2, z_3) and then to scaled perimetric coordinates, $(u, v, w) \equiv (Az_1, Bz_2, Cz_3)$.
3. Apply the operators to the general wavefunction basis $e^{-\frac{1}{2}(Az_1+Bz_2+Cz_3)}L_l(Az_1)L_m(Bz_2)L_n(Cz_3)$.
4. Use the Laguerre recurrence relations to eliminate all partial derivatives and powers of the coordinates.
5. Save the 57 term recursion relation (33 term for fixed nucleus) to a text file for use by other codes.

Minimal working examples are now provided to expand on these points.

Maple: Define operators

```
1 # Operators defined in full and converted into atomic units
  ↪ later
2 # The kinetic energy operator
```

```

3 T:=psi->-hbar^2/2*(Sum(1/(mu[i]*r[i]^2)*Diff(r[i]^2*
  ↪ Diff(psi,r[i]),r[i])+Sum(b[i,j]/(mu[i,j])*
  ↪ Diff(psi,r[i],r[j]),j=(i+1)..3),i=1..3)):
4
5 # The potential energy operator
6 V:=psi->e^2/(4*Pi*epsilon[0])*Sum(Sum(Z[i]*Z[j]/r[i,j],
  ↪ j=(i+1)..3),i=1..3)*psi:
7
8 # The Hamiltonian operator
9 H:=psi->T(psi)+V(psi):
10
11 # The overlap
12 S:=psi->E*psi:
13
14 #Apply atomic units:
15 e:=-1: #The elementary charge is -1
16 hbar:=1: #The reduced planks constant is 1
17
18 k:=1/(4*Pi*epsilon[0])=1: #Coulomb's constant is 1
19 epsilon[0]:=solve(k,'epsilon[0]'):
20
21 #Apply mass conversions
22 mu[1] := 1/(1/m[1] + 1/m[3]):
23 mu[2] := 1/(1/m[2] + 1/m[3]):
24 mu[3] := 1/(1/m[1] + 1/m[2]):
25
26 mu[1,2] := m[3]:
27 mu[1,3] := m[1]:
28 mu[2,3] := m[2]:
29
30 b[1,2]:=(r[1]^2+r[2]^2-r[3]^2)/(r[1]*r[2]):
31 b[1,3]:=(r[1]^2+r[3]^2-r[2]^2)/(r[1]*r[3]):
32 b[2,3]:=(r[2]^2+r[3]^2-r[1]^2)/(r[2]*r[3]):

```

Now to transform inter-particle coordinates, r_i to perimetric coordinates, z_i .

Maple: Coordinate transformation to perimetrics

```

1 # Coordinate transforms require the forward transformations
  ↪ and optionally the reverse transformation.
2
3 R1:=(z[2]+z[3])/2:#Forward transformation
4 R2:=(z[1]+z[3])/2:
5 R3:=(z[1]+z[2])/2:
6 tr:={r[1]=R1,r[2]=R2,r[3]=R3}:
7 rR1:=-r[1]+r[2]+r[3]:#Reverse transformation
8 rR2:=r[1]-r[2]+r[3]:
9 rR3:=r[1]+r[2]-r[3]:
10 rtr:={z[1]=rR1,z[2]=rR2,z[3]=rR3}:
11 nv:={z[1],z[2],z[3]}:#New variable names
12
13 Jac:=Student['MultivariateCalculus']['Jacobian']([R1,R2,R3],
  ↪ [z[1],z[2],z[3]],output=determinant);
14
15 g:=r[1]*r[2]*r[3]*8*Pi^2: #This is the volume element of
  ↪ Cartesian to triangular coordinate change.
16
17 KE:=psi->T(psi):
18 PE:=psi->V(psi):
19 OV:=psi->S(psi):
20
21 wavefn:=psi(u,v,w): #The wavefunction is left undefined for
  ↪ now
22
23 KE:=expand(simplify(PDEtools[dchange](tr,(KE(wavefn)*g*Jac),nv,
  ↪ rtr))):
24 PE:=expand(simplify(PDEtools[dchange](tr,(PE(wavefn)*g*Jac),nv,
  ↪ rtr))):
25 OV:=expand(simplify(PDEtools[dchange](tr,(OV(wavefn)*g*Jac),nv,
  ↪ rtr))):

```

Now to transform from perimetric to scaled perimetric coordinates.

Maple: Coordinate transformation to scaled perimetrics

```

1 U:=A*z[1]:
2 V:=B*z[2]:
3 W:=C*z[3]:
4 NLP:={A,B,C}:
5 tr:={z[1]=solve(u=U,z[1]), z[2]=solve(v=V,z[2]),
  ↪ z[3]=solve(w=W,z[3])}:
6 nv:={u,v,w}:
7
8 Jac:=Student['MultivariateCalculus']['Jacobian']([
  ↪ solve(u=U,z[1]),
  ↪ solve(v=V,z[2]),solve(w=W,z[3])],[u,v,w],
  ↪ output=determinant);
9 KE:=eval(expand(simplify(PDEtools[dchange](tr,KE*Jac,nv,
  ↪ params=NLP)))):
10 PE:=eval(expand(simplify(PDEtools[dchange](tr,PE*Jac,nv,
  ↪ params=NLP)))):
11 OV:=eval(expand(simplify(PDEtools[dchange](tr,OV*Jac,nv,
  ↪ params=NLP)))):

```

Now the operators have been defined, the wavefunction form can be specified. The Laguerre polynomials are not needed at this step, thus the wavefunction is left in a general form, $F_{l,m,n}(u,v,w)$.

Maple: State wavefunction form

```

1 psi(u,v,w):=exp(-u/2-v/2-w/2)*F[l,m,n](u,v,w):
2 KE:=expand(simplify(calculate(KE))):
3 PE:=expand(simplify(calculate(PE))):
4 OV:=expand(simplify(calculate(OV))):

```

The C++ markers are now introduced, which are used by 3Body to filter out terms relevant to each of the operators in order to build the relevant matrices.

- **hh:** For left hand side of eigenproblem.
- **ss:** For right hand side of eigenproblem.
- **t:** For kinetic energy terms.
- **p:** For potential energy terms.

- **o**: For overlap terms.

A separate Maple code was written which processes the terms in the Schrödinger equation called `RecursionRelations.mpl` and is called using `Apply RecursionRelations()`. This code is complex so will not be discussed in detail here, but an overview of the steps are as follows:

1. Search through terms in the Schrödinger equation and looks for the terms with derivatives or powers of u, v, w .
2. Apply the relevant Laguerre recurrence relation, equation (2.20) to simplify the term.
3. Output the processed term and then save to file.

The `RecursionRelations.mpl` is also used to build the recursion relations for all the other operators such as $\hat{r}_i^3, \delta(\hat{r}_i), \hat{\tau}_i$ which are used by the Expectation C++ code. The Schrödinger equation is constructed using the necessary operators and it can be processed by a single function call. The output text file will have the following format.

Bash: RR format

```
1 |[dl,dm,dn],RecursionRelation
```

where `dl, dm, dn` represent the integer l, m, n subscripts and `RecursionRelation` represents the corresponding coefficient for that triple index.

Maple: Applying The Recursion Relations

```
1 # Build the terms involved in the Schroedinger equation.
  ↳ Note the sign of the overlap is dealt with in the C++
  ↳ program
2 Expr:=simplify(KE*t*hh+PE*p*hh+OV*o*ss,size):
3
4 read "RecursionRelations.mpl" # Where the recursion relation
  ↳ code is stored
5 ApplyRecursionRelations(Expr) # Call the function to apply
  ↳ the recursion relations
```

The final step is to save the recursion relations to file

Maple: Save recursion relations

```

1 SaveRRToFile:=proc(RR,FileName)
2   local Fterms, size, index, dl, dm, dn, Expr, fd:
3   Fterms := (indets(RR,function)):
4   size:=nops(Fterms):
5   fd:=fopen(FileName, WRITE, TEXT):
6   for index from 1 to size do
7     dl:=eval(op([0,1],Fterms[index]),[l=0,m=0,n=0]):
8     dm:=eval(op([0,2],Fterms[index]),[l=0,m=0,n=0]):
9     dn:=eval(op([0,3],Fterms[index]),[l=0,m=0,n=0]):
10    Expr:=coeff(RR,Fterms[index]):
11    fprintf(fd,"| [%A,%A,%A],%A",dl,dm,dn,Expr):
12  end do:
13  fclose(fd):
14 end proc:
15
16 #ABC wavefunction
17 Transformation:=[]:
18 SaveRRToFile(expand(simplify(eval(RR,Transformation))),
19   ↪ "ABC/RR.rr"):
20
21 #AC wavefunction
22 Transformation:=[B=A]:
23 SaveRRToFile(expand(simplify(eval(RR,Transformation))),
24   ↪ "AC/RR.rr"):
25
26 # etc... for the different wavefunction types (PEKERIS, K,
27   ↪ A, AC, ABC)

```

3.5.2 Inter-Particle Cusps

Section 2.6 discusses the use of inter-particle cusp conditions as a quality control test for the wavefunctions produced in this work. The wavefunctions are read from file and the cusps calculated using the form given in equation (2.42). This is done using the following Maple code which uses the inbuilt `int` command of Maple. The following example calculates the ν_{21} cusp value.

Maple: Calculating Cusps in Maple

```

1 restart:
2 Digits:=32: #Specify quadruple precision
3
4 u:=z1*A: v:=z2*B: w:=z3*C: #Convert scaled perimetric
   ↪ coordinates to perimetric coordinates
5 z1:=r2+r3-r1: z2:=r1+r3-r2: z3:=r1+r2-r3: #Convert
   ↪ perimetric coordinates to inter-particle coordinates
6 filename:="SAVEVECTOR": #Set filename here
7 if FileTools[Exists](filename) then
8     mempi:=evalf(Pi):unprotect('Pi'):
9     read filename: # Load filename
10    Pi:=mempi:protect('Pi'):
11
12    ##Detect wavefunction type from input file
13    if (C = 'C') then
14        C:=A+B: #AB psi
15    fi:
16    if (B = 'B') then
17        B:=A: #A
18    fi:
19    if (A = 'A') then
20        A:=ep: #PEKERIS and K
21    fi:
22
23    #If denominator of cusp expression has been calculated in
   ↪ logfile, don't calculate again and take from log file.
   ↪ This option is filled in by the bash script CalcCusps
24    DR3VAL:=DR3VAL:
25
26    g:=4*Pi*r1*r2: # Jacobian of transformation. Integrating
   ↪ over full volume element,  $8\pi^2 r_1 r_2 r_3$  is not
   ↪ required as  $r_3$  is set to zero for coalescence of
   ↪ particles 1 and 2
27    L:=LaguerreL: #Tell Maple, L in filename is a Laguerre
   ↪ polynomial
28    st:=time(): #Start timer
29

```

```

30  psi:=expand(simplify(exp(-u/2-v/2-w/2)*(F))): #Multiply
    ↪ wavefunction basis (F) by exponential parameter
31
32  dpsi:=simplify(diff(psi,r3)): #Differentiate wavefunction
33  v21:=int(eval(simplify(eval(psi*dpsi*g,[r2=r1])),r3=0),
    ↪ r1=0..infinity): #Evaluate the v21 cusp
34  printf("numerator: %A\n",v21):
35  #If denominator cannot be found in log file, calculate
36  if DR3VAL=1 then
37      dr3:=int(eval(simplify(eval(psi*psi*g,[r2=r1])),r3=0),
    ↪ r1=0..infinity):
38  else: #Take value from log file
39      dr3:=DR3VAL:
40  fi:
41  #Now combine numerator and denominator and print result to
    ↪ terminal
42  printf("denominator: %A\n",dr3):
43  v21cusp:=v21()/dr3():
44  printf("v21 cusp value: %A\n",v21 cusp):
45  st:=time() -st:
46  printf("the calculation took: %A seconds\n",st):
47  else
48  printf("Error %a does not exist",filename): #Error if file
    ↪ doesn't exist
49  error "File not found"
50  fi:

```

In Section 2.6 two methods were discussed for calculating the inter-particle cusps, the series expansion method and the gradient method. The gradient method is more accurate than the expansion method introduced by Pack and Byer's Brown^[58] with examples of nucleus-electron (ν_{31}) and electron-electron (ν_{21}) cusps now shown for an atomic system with fixed nucleus, H^- ($e^-e^-\infty^+$) and a molecular system $\text{H}_2^+(p^+p^+e^-)$.

Table 3.7 Cusps calculated using the expansion method and gradient method for fixed nucleus helium and the hydrogen molecular ion. A wavefunction size of 2856 was used for both systems at an accuracy of double precision.

System	$\langle X \rangle$	Expansion Method	Gradient Method
$e^-e^-\infty^+$	ν_{31}	-1.995 328 909	-1.999 999 91
	ν_{31}^{exact}	-2	-2
	ν_{21}	0.492 866 400	0.499 993
	ν_{21}^{exact}	0.5	0.5
$p^+p^+e^-$	ν_{31}	0.101 139 403	-0.999 456 025
	ν_{31}^{exact}	-0.999 455 679	-0.999 455 679

ν_{31}^{exact} and ν_{21}^{exact} are the exact cusp ratios calculated using the Kato cusp condition.

The accuracy of the cusp calculations for both atomic and molecular systems is greatly improved with the use of the gradient method over the expansion method. The ν_{21} cusp is not included for H_2^+ due to known inaccuracies in the expectation value of the proton-proton delta function^[35], $\langle \delta(r_{12}) \rangle$, and more generally with systems where $m_1, m_2 \gg m_3$. These inaccuracies warrant further research but have been suggested to be caused by wavefunctions which cannot correctly account for the nucleus-nucleus contact properties. This may require much larger basis sets to be employed or different functions used within the basis.^[35]

3.5.3 Numerical Integration

Section 3.4.3 outlined how particle densities are calculated using C++. The key part of this process is Maple building the integrand and converting into a C++ recognisable format. A minimal working example is now given showing the integrand building process for the intracule density form given in equation (3.5).

Maple: Building the integrand for intracule density

```

1 restart:
2 Digits:=32:
3
4 L:=LaguerreL:
5
6 # Load the wavefunction from file

```

```

7 psi:=cat(kernelopts(homedir),"/../path_to_wavefunction_file
  ↪ .sv"):
8
9 read psi:
10
11
12 unassign('u','v','w','z1','z2','z3','r1','r2','r3'):
13 CC:=C: # The letter C clashes with the CodeGeneration
  ↪ library, so is switched for CC
14
15 # Introduce NLPs
16 psi:=eval(psi,[u=A*z1,v=B*z2,w=CC*z3]):
17
18 # Convert to inter-particle coordinates
19 z1:=r2+r3-r1:
20 z2:=r1+r3-r2:
21 z3:=r2+r1-r3:
22
23 with(CodeGeneration):
24
25 ### Coordinate transformation from inter-particle to
  ↪ perimetric form
26
27 r1:=(v/B+w/CC)/2:
28 r2:=(u/A+w/CC)/2:
29 r3:=(v/B+u/A)/2:
30
31 # Specify Jacobian factor required for integrating over
  ↪ Dirac delta function. Due to the sifting theorem of the
  ↪ Dirac delta function, r3 has been integrated out at the
  ↪ beginning and will be the argument of the dirac delta
  ↪ function
32 gDirac:=4*Pi*r1*r2/(2*r3):
33
34 # Jacobian for u->ri conversion
35 gDirac:=gDirac/(2*CC*A):
36

```



```

37 # Limitt is a large number substituted in place of infinity
   ↳ required to transform the integration limits to that of
   ↳ a unit-hypercube. If infinity was desired then the
   ↳ program would be trickier as would need to be handled as
   ↳ a limit problem
38
39 limitt:=10000:
40 Op:=4*Pi*r3^2: # Spherical average if normalised intracule
   ↳ density is desired
41
42 # eta represents the dummy variable used for the
   ↳ substitution. x and y are used as integration variables
   ↳ in the C++ script.
43
44 # u is eliminated by the substitution
45 u:=2*A*r3val*eta:
46 # Introduce dummy variable
47 eta:=x:
48 w:=limitt*y:
49
50 # EXPANDED represents the integrand which also has a
   ↳ simplification applied to reduce storage size of the
   ↳ integrand
51 EXPANDED:=simplify(limitt*2*A*r3val*psi*Op*psi*gDirac,
   ↳ LaguerreL):
52 EXPANDED:=eval(EXPANDED,v=2*A*r3val-u):
53
54 unassign('r'):
55
56 # Create and open a file to save the integrand
57 shortname:=FC_DR3D:
58 SVFileName:=(cat(FileTools[Basename](SVFileName),"_",shortname,
   ↳ ".txt")):
59
60
61 fd := fopen(SVFileName, WRITE, TEXT):
62

```

```

63 # This part uses a regex to convert Maple maths commands to
    ↪ ones recognised by the C++ CUBA library. This list can
    ↪ be expanded for other integrands which contain other
    ↪ operations
64 StringTools:-RegSubs ("^[^=]*=","=",
    ↪ C(EXPANDED, 'output=string')) :
65 StringTools:-SubstituteAll(%,"pow","powl"):
66 StringTools:-SubstituteAll(%,"sqrt","sqrtl"):
67 StringTools:-SubstituteAll(%,"exp","expl"):
68
69 # Print the integrand to the file in a string format
70 fprintf(fd,"%s", %);
71
72 # Close and save the integrand file
73 fclose(fd):

```

The file is now ready to be read by the C++ code to calculate particle densities or expectation values.

3.5.4 Excited State Matrices

The matrices for the excited state problem are built in Maple and then saved to file due to their larger byte sizes as outlined later in Section 3.6. The 57-term recursion relation with all parameters left undefined, A , B , C , m_i , Z_i , produced and saved by Maple is then read back into Maple and the matrices built using the mathematics outlined in Chapter 6. Due to all parameters being left undefined in our methods, technically the excited states matrices \mathbf{H} and \mathbf{S} only need to be generated once, although Section 3.6 discusses the memory issues with doing this for the excited state method.

The 57-term recursion relation contains markers hh, ss, t, p, o discussed in Section 3.4.1 which allow for specific sub-matrices to be extracted from the main matrix. Firstly the recursion relation is read from file.

Maple: Read in and load A, B, C recursion relation from file

```

1 restart:
2
3 with(StringTools): #Load StringTools library for later
4 with(LinearAlgebra):

```

```

5 with(MTM): #Load MTM library used for extracting lower
  ↳ triangular matrices
6 Triplet:=True: #Specifies if Singlet or Triplet matrices are
  ↳ to be generated
7 omega:=20: #Set size of matrix using maximum matrix order
8
9 st:=time(): #Setup timer
10 sst:=st:
11
12 #Load Recursion relations. Specify directory location
13 dirname:=cat(kernelopts(homedir),"/Dropbox/3body/Maple/SE
  ↳ /RecursionRelations"):
14
15 if FileTools[Exists](dirname) = false then
16   error("Recursion Relations not found"):
17 else
18   cwd:=currentdir():
19   currentdir(dirname):
20   read "SE_RR.m": #Load the recursion relations
21   currentdir(cwd):
22 fi:
23
24 #This maps the recursion relations by index for easy lookup
25 RR_Map:=table(sparse):#Sparse fill with zeros for ones we
  ↳ don't provide a value
26 RR_Indices:=indets(RR,function):#Finds all the
  ↳ F[l,m,n](u,v,w)
27 RR_Indices:=map2(op,0,RR_Indices):#Strips the (u,v,w)
28 RR_Indices:=select(type,RR_Indices,indexed):#Makes sure we
  ↳ only have elements with indices
29
30 ## Normal coefficients
31 for RR_Index in RR_Indices do
32   Index:=eval(op(RR_Index),[l=0,m=0,n=0]):
33   RR_Map[Index]:=simplify(coeff(RR,RR_Index(u,v,w))):
34 od:

```

`RR_Map[Index]` acts as a lookup table by taking the triple index $\{l, m, n\}$ and returning its coefficient. Next the maximum number of terms is calculated from

the ω provided at the start of the code, then the asymmetric numbering scheme is implemented.

Maple: Asymmetric numbering scheme

```

1 #Generates the k index from the l,m & n index.
2 GenerateSingleIndexFromTriple[Asymmetric]:=proc(l,m,n)
3   local f:
4   f:=proc(c,r)
5     binomial(c+r-1,r);
6   end proc:
7   f(l+m+n,3)+f(l+m,2)+f(l,1)+1;
8 end proc:
9
10 #Checks to see if we have already calculated this k index
    ↳ (not necessary but included just in case)
11 ListHasIndex:=proc(List,Index)
12   return
    ↳ member(Index,convert({indices(List,'nolist')},list)):
13 end proc:
14
15 LMN_Map[Asymmetric]:=table(): #This is the asymmetric rule
    ↳ to form the asymmetric matrix.
16 for ww from 0 to omega do
17   for vv from 0 to ww do
18     for uu from 0 to vv do
19       l:=uu:
20       m:=vv-uu:
21       n:=ww-vv:
22
23       ↳ tval:=GenerateSingleIndexFromTriple[Asymmetric](l,m,n):
24       if ListHasIndex(LMN_Map,tval) = false and
25         ↳ (l+m+n)<=omega then
26         LMN_Map[Asymmetric][tval]:=[l,m,n]:
27       fi:
28     od:
29   od:
30 od:
31 l:='1':#Unassign these otherwise they conflict with the RR

```

```

30 m:='m':
31 n:='n':

```

To build the excited state matrix the triple index $\{l, m, n\}$ is assigned a single integer index i or k which denote the rows and columns of the matrix (discussed in Section 2.2.5). Chapter 6 discusses how these matrices are generated as the original ground state method + the corrective elements required to allow A and B to vary independently. Firstly the ground state matrix is generated, then the nested double “for” loop (jj, ii) applies the additional $\xi\delta(n, n')$ condition to each matrix element.

Maple: Building excited state matrices in Maple

```

1 Size1:=nops(convert({indices(LMN_Map[Asymmetric])},list)):
2 'Full_Asymmetric_Matrix_Size'=Size1;
3
4 B_Matrix:=Matrix(Size1,Size1,0): #Creates a matrix of zeros
   ↪ to append the asymmetric elements to
5
6 #Setup timer
7 allst:=time():
8
9 print("Commencing building of matrices"):
10
11 #These values are all left undefined, or modified if user
   ↪ wants to generate less general matrices (e.g. for atomic
   ↪ systems)
12 Transformation:=[l=L,m=M,n=N,hh=1,E=1,m[1]=m1,m[2]=m2,m[3]=m3
   ↪ ,Z[1]=z1,Z[2]=z2,Z[3]=z3,o=1]:
13
14 #I and J are traditional indices for the columns and rows of
   ↪ a matrix however in Maple "I" is protected as the
   ↪ imaginary number, so using H instead.
15 for H from 1 to Size1 do
16   for J from 1 to Size1 do
17     L:=LMN_Map[Asymmetric][H][1]:
18     M:=LMN_Map[Asymmetric][H][2]:
19     N:=LMN_Map[Asymmetric][H][3]:
20     Lp:=LMN_Map[Asymmetric][J][1]:
21     Mp:=LMN_Map[Asymmetric][J][2]:

```

```

22  Np:=LMN_Map[Asymmetric][J][3]:
23  lambda:=Lp-L:
24  mu:=Mp-M:
25  nu:=Np-N:
26
27  ↪ B_Matrix[H,J]:=eval(RR_Map[lambda,mu,nu],Transformation):
new_factor2:=(WW,XX,YY,ZZ)->((-1)^(XX+ZZ))*
28  ↪ ((4*A*B)/(A+B)^2)*((B-A)/(B+A)^(WW+YY+XX+ZZ))*
29  ↪ hypergeom([-YY,-XX],[1],-(4*A*B/((B-A)^2)))*
30  ↪ hypergeom([-WW,-ZZ],[1],-(4*A*B/((A-B)^2))):
31  B_Matrix1[H,J]:=0:
32  for jj from -2 to 2 do
33    for ii from -2 to 2 do
34      ↪ B_Matrix1[H,J]:=eval((B_Matrix1[H,J]+(RR_Map[ii,jj,nu]*
35      ↪ new_factor2(L+ii,M+jj,Lp,Mp))),Transformation):
36    od:
37  od:
38  gc():
39  od:

```

Maple stores the algebraic matrix elements in the kernel memory and they are extracted to be saved to their respective files. The dimensions of the matrix are not carried through and instead each matrix element is put onto a separate line in the file with only the lower triangular part of the matrix being saved to reduce file size by exploiting the matrix symmetry. The code snippet above will build the entire matrix; so sub matrices are now extracted using the markers within the matrix, *hh*, *ss*, *t*, *p*, *o*. A highly optimised example on how to extract the kinetic energy matrix, **T** is now given. It extracts this sub matrix as a coefficient of the kinetic energy marker, *t*. Use of the tilde operator \sim in Maple applies the action to the whole matrix, which makes this one of the most efficient methods to conduct this operation; and is much faster and memory efficient than using nested procedures to loop through each matrix element.

Maple: Extracting a sub matrix and save to file

```

1  #B_Matrix represents the ground state matrix
2  #B_Matrix1 represents the additional excited state
   ↪  mathematics
3
4  #Specify the sign of the overall wavefunction for generating
   ↪  either singlet or triplet matrices
5  if Triplet=True then
6    BB_Matrix:=B_Matrix-B_Matrix1: # Triplet -
7  else
8    BB_Matrix:=B_Matrix+B_Matrix1: # Singlet +
9  fi:
10
11 # This extracts the kinetic energy terms and simplifies its
   ↪  byte-size
12 TT:=Matrix(map(simplify,coeff~(BB_Matrix, t),size),
   ↪  shape=triangular[lower]):
13
14 #Save the kinetic energy matrix to file
15 print("Saving kinetic energy matrix to file"):
16
17 #Create and open empty file
18 filenameTT:=fopen(cat("Matrix_SE_ASYM", Size[Asymmetric],
   ↪  "x",
   ↪  Size[Asymmetric],"TT_atom_triplet.txt"),WRITE,TEXT):
19
20 #Loop over matrix and save
21 nRow,nCol :=LinearAlgebra[Dimension](TT):
22 for i from 1 to nRow do
23   for j from 1 to nCol do
24     TT_str:=convert(TT[i,j],string):
25     fprintf(filenameTT,"%s\n", TT_str):
26   od:
27 od:
28 fclose(filenameTT):

```

3.6 Python

Python was primarily used to implement and solve the excited state problem. A detailed account of the theory is given in Chapter 6, whilst the details of the implementation will be described in this section. Python was also used to develop scripts for data analysis and plotting but these will not be discussed here.

3.6.1 Choice of Python

C, C++, Fortran and various other examples are known to be very fast and well implemented programming languages so why use Python? Usually when choosing the Python language you sacrifice performance for productivity and ease of use due to it being an interpreted language rather than a compiled one⁸. Writing an eigenvalue solver in pure Python would normally be substantially slower than one written in a lower level language; however Python has many advantages that allows one to bypass this issue, and several will now be discussed.

3.6.1.1 Language flexibility

There are a wide variety of programming languages currently available with more in development. Python was not created to suit a specific need and therefore is well-suited to quick development across a wide array of applications. Due to this versatility it has firm support from programmers across disciplines and thus a vast selection of well optimised libraries are available to assist with problems shortening development time. The language is object-oriented (OO) and intuitive which makes it easy to debug or modify.

3.6.1.2 Language Interoperability

Something that Python excels at is “gluing” other programming languages together. It can be used to call R using RPy, Fortran using F2py, C using Cython etc... A Python library which uses this functionality is SciPy. SciPy is a python-based open-source coding library designed for mathematics, science and engineering. A lot of SciPy acts as a bridge to scientific routines that are freely available at <http://www.netlib.org/>. Netlib is a large repository of useful and robust scientific algorithms written in very fast C, C++ and Fortran. NumPy and SciPy use highly optimised C and Fortran implementations from various libraries such as Automatically Tuned Linear Algebra Software (ATLAS), Basic Linear Algebra Subprograms (BLAS)

⁸It is however possible to compile Python codes to make them faster using libraries such as Cython.

and Linear Algebra PACKage (LAPACK) which are the gold standard of linear algebra vector and matrix computations. They are in-fact used as a benchmark speed test for scientific computing hardware, so Intel and other CPU developers work hard at updating and optimising these routines.

SciPy does not reinvent the wheel by reprogramming these routines but generates “wrappers” to them so these algorithms can be used in Python. The method of wrapping these routines has a minimal effect on the original speed of the algorithms, which is now demonstrated by comparison between a Python wrapper of OpenBlas vs. the C implementation of OpenBlas.

A Python and C code were written which both use the C implementation of the matrix dot product function of OpenBlas (The Python code wraps this function and C directly calls it). Both codes generate two equal sized random matrices, then their dot product is calculated along with the required CPU time. The time results vs. matrix size have been plotted in Figure 3.3.

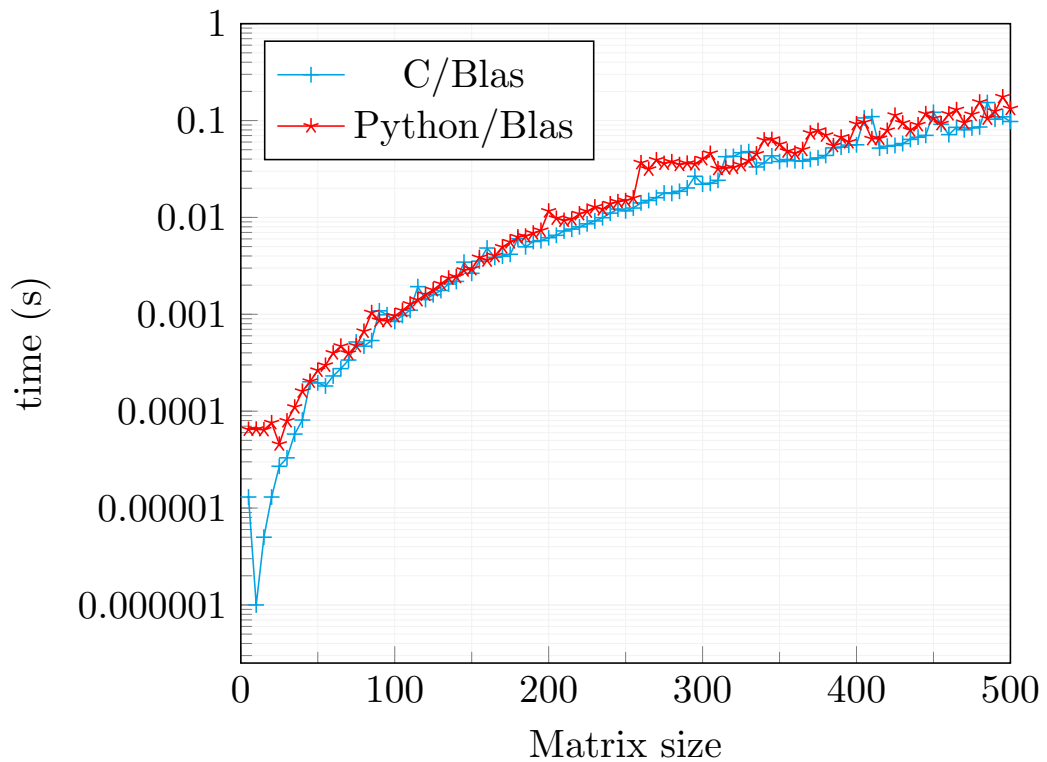


Figure 3.3 Python wrapper vs. C implementation of the same OpenBlas dot product function. Note, the y-axis is plotted on a log scale.

Figure 3.3 shows the timings of both the Python wrapper and C codes are almost identical across matrix sizes, although a difference is seen for very small matrix sizes

where the time required to pass the input from Python to C is much greater than the calculation time itself.

3.6.2 Excited state code development

Chapter 6 discusses the theory and results for the excited state problem implemented and extended based on the work of Pekeris^[50]. The modification in the theory results in a dramatic increase in the computational complexity of the problem caused by the following points which will be discussed along with solutions to circumvent the difficulties:

1. Sparsity of secular determinant is lost by addition of a “filtered” recursion relation representing $\delta(n, n')$ which also causes a substantial increase in non-zero matrix element byte sizes.
2. Use of the asymmetric numbering scheme as no wavefunction symmetry can be exploited, meaning larger matrices are required.
3. Ill-conditioned generalised eigenvalue problem.

3.6.2.1 Loss of sparsity

Within the ground state problem, the Laguerre orthogonality conditions for the triple indices l, m, n results in sparse matrix representations for the Hamiltonian and overlap matrices **H** and **S**. This provides a computational advantage, however this sparsity is lost when the required modifications to the theory are applied in order to model highly excited states. Figure 3.4 shows plots of **H** for the ground state and excited state methodologies.

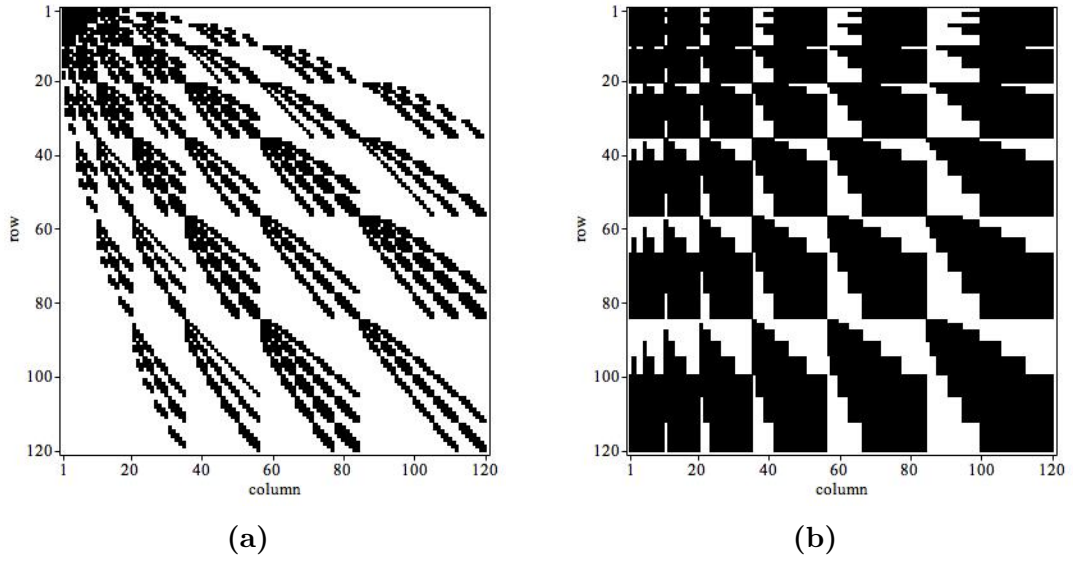


Figure 3.4 Sparsity comparison between \mathbf{H} for (a) the ground state and (b) excited state methodologies. Black squares represent non-zero matrix elements and white squares represent zero matrix elements.

Figure 3.4 (a) has a 77.09% sparsity, whereas Figure 3.4 (b) has a 29.12% sparsity. As the secular determinant is now dense, the choice of algorithm used to solve the generalised eigenvalue problem is important. The DSYEVD routine from LAPACK is a good candidate as it is designed to handle dense matrices and has the ability to only calculate eigenvalues (using the Pal-Walker-Kahan variant of the QL algorithm^[73]). This is of benefit in the current work as the eigenvectors are not required for each optimisation step, and only need calculating at the end once the optimised NLP values have been determined. Once the optimisation routine is complete and a minimised eigenvalue is obtained, the eigenvector is calculated using a divide and conquer algorithm. This sequence of events is much faster than calculating all the eigenvalues and eigenvectors for every optimisation step.

3.6.2.2 Asymmetric numbering scheme

When correctly accounting for fermionic exchange in the ground state methodology, the symmetric numbering scheme can be exploited for singlet states and the antisymmetric numbering scheme for triplet states, which reduces the size of the required matrix, simplifying the problem. Due to the different way of accounting for fermionic exchange in the excited state method the asymmetric numbering scheme must be used (discussed in Section 2.2.5) which requires significantly larger matrices to reach the same matrix order ω for the equivalent combinations of $\{l, m, n\}$ required. As an example, for $\omega = 30$ using the symmetric numbering scheme results in a 2856

size matrix whereas using the asymmetric numbering scheme results in a 5456 size matrix.

These generated generalised matrices which represent the secular determinant are much larger than those used for the ground state problem. With all parameters left undefined (m_i, Z_i, A, B, C) the resulting matrix to store in memory or save to file is of the order of gigabytes. Table 3.8 shows a comparison of the required storage size for the ground state and excited state methods for the lower triangular Hamiltonian matrix, \mathbf{H} . Due to all matrices being symmetric in both ground and excited state methods, only the lower triangular matrix is required for the DSYEVD algorithm and thus the required storage size is almost halved.

Table 3.8 Comparison of byte size for the lower triangular matrix representing \mathbf{H} using the ground state and excited state methodologies for various values of $\omega = l + m + n$. All m_i, z_i and NLPs left in their algebraic form.

	\mathbf{H} , matrix byte size (Mega Bytes)			
ω	Matrix size (G.s)	Ground state (G.s)	Matrix size (E.s)	Excited state (E.s)
5	34	0.04	56	2.62
10	161	0.41	286	68.72
15	444	1.78	816	595.92
20	946	5.97	1771	3017.13
25	1729	16.63	3276	11126.44
30	2856	40.68	5456	24478.10

The lower triangular, excited state Hamiltonian matrix, \mathbf{H} with matrix order $\omega = 30$ is ≈ 2 times larger in matrix size but ≈ 600 times larger (byte-size) than the equivalent order ground state matrix. All the matrix byte sizes presented in Table 3.8 are the result of already simplifying each non-zero matrix element in Maple; so this highlights how computationally demanding this method is compared to the ground state problem.

3.6.2.3 Ill conditioned eigenvalue problem

In Section 3.6.2.1 a wrapper to the DSYEVD algorithm was discussed; but this was found to not always be numerically stable when the matrix size becomes larger or the A and B NLPs are too similar. This is still the default algorithm used; however more stable methods were explored for solving these ill defined cases such as the calculation for lower lying excited singlet states. The condition number (cond) of the Hamiltonian and overlap matrices \mathbf{H} and \mathbf{S} were used as a measure of the numerical

stability of the generalised eigenvalue problem, which is defined using a general matrix \mathbf{A} as^[101]

$$\text{cond}(\mathbf{A}) = \|\mathbf{A}\| \cdot \|\mathbf{A}^{-1}\| = \frac{\lambda_h}{\lambda_l} \quad (3.10)$$

Where λ_h, λ_l represent the highest and lowest eigenvalues respectively. As the matrix sizes increase or the A and B NLPs become too similar in the excited state problem, the condition number becomes larger, and when approaching a matrix size of $\approx 2000 \times 2000$ it is significantly higher than the inverse of the machine epsilon which at standard double precision is 1×10^{16} . It was found that the ground state method also suffers from high matrix condition numbers, but these never exceeded the inverse of the machine epsilon; whereas for the excited state matrices the condition number can readily exceed 1×10^{30} . This makes any matrix inversions subject to substantial floating point rounding errors meaning standard routines such as Cholesky decomposition can fail as the overlap matrix ends up being negative definite which is not permissible. The quick way to bypass this problem would be to run the calculations at higher precision but this would be *significantly* slower; and will not necessarily resolve the issue hence trying other means are advisable. Various alternatives were explored including LU decomposition^[72], LQ^[72] decomposition, repeated applications of Singular Value Decomposition (SVD)^[101] etc... but these either suffered from numerical stability issues or were too slow. However, a method that offered substantial improvement was the QZ algorithm. The generalized eigenvalue problem consisting of matrices \mathbf{X} and \mathbf{Y} , with λ representing the eigenvalues and ν the eigenvectors is

$$\mathbf{X}\nu = \lambda\mathbf{Y}\nu \quad (3.11)$$

The QZ decomposition, also known as the generalised Schur decomposition factorises both matrices as^[102]

$$\begin{aligned} \mathbf{X} &= \mathbf{Q}\mathbf{R}\mathbf{Z}^T \\ \mathbf{Y} &= \mathbf{Q}\mathbf{S}\mathbf{Z}^T \end{aligned} \quad (3.12)$$

where \mathbf{Q}, \mathbf{Z} are unitary and \mathbf{R}, \mathbf{S} are upper triangular and represent the Schur forms of \mathbf{X} and \mathbf{Y} respectively. The QZ algorithm applied to a matrix pair (\mathbf{X}, \mathbf{Y}) is equivalent to applying the QR algorithm on $\mathbf{X}\mathbf{Y}^{-1}$ and $\mathbf{Y}^{-1}\mathbf{X}$ at the same time with the added advantage of no matrix inversion being required, and is thus less affected by the condition number of matrix \mathbf{Y} ^[102]. The eigenvalues λ that solve

the generalised eigenvalue problem in equation (3.11) are calculated as the diagonal elements of the ratio of matrices \mathbf{R} and \mathbf{S} in equation (3.12)

$$\text{diag}\left(\frac{R_{ii}}{S_{ii}}\right) = \lambda_i. \quad (3.13)$$

This process is slightly slower than using the DSYEVD algorithm however is substantially more stable to numerical errors. Implementation of this was done by using the SciPy wrapper to the LAPACK functions `gges` and `tgssn`; with an important point being *not* to compile LAPACK against the Apple accelerate framework due to pronounced numerical instability issues. In order to obtain the eigenfunctions from the QZ decomposition, the eigenvalue, λ_i is substituted into equation (3.11) and the null space is calculated, i.e. the matrix, $\boldsymbol{\nu} = \text{null}(\mathbf{X} - \lambda_i \mathbf{Y})$

$$(\mathbf{X} - \lambda_i \mathbf{Y})\boldsymbol{\nu} = 0. \quad (3.14)$$

Two methods have been implemented, the first being Gaussian elimination followed by back-substitution^[72], which unfortunately suffers from numerical instability issues, so the second, default method performs a Singular Value Decomposition (SVD)^[101] to obtain the generalised eigenvector, $\boldsymbol{\nu}$.

Balancing the matrices was found to increase the numerical stability further, which in this work is done by dividing the matrix elements in each row by the element in the row which sits on the diagonal of the matrix making all the diagonal elements 1. Both the DSYEVD and QZ methods along with matrix balancing can be specified within the excited state run file for ease of use. This run file can be seen in Appendix A.

Future work would see an implementation of generalised singular value decomposition which should theoretically be one of the most numerically stable ways to solve a generalised eigenvalue problem^[101]. This was attempted, however wrapping the LAPACK function GSVD in Python proved difficult.

3.6.3 Processing matrices

Solutions are now discussed to circumvent the computational difficulties outlined in the previous section. In the ground state C++ code the matrices are calculated within C++ using the Maple recursion relations. This can be done as the ground state matrices are small with regards to byte size and hence can easily be loaded into RAM and held throughout the calculation. This is not possible with the much larger byte size excited state matrices using the available hardware. The excited state matrices that represent \mathbf{H} and \mathbf{S} are calculated and simplified in Maple and then saved to file.

This is required as they can be read from disk without being explicitly loaded into memory during the calculation which will overload most non-HPC machines.

Four methods have been developed to simplify processing of the excited state matrices:

1. Build the abstract syntax tree (AST) for each arithmetic matrix element and serialise it into fast, efficient run-time code.
2. Evaluate the arithmetic impromptu.
3. Generate less general matrices.
4. A combination of methods 1, 2 and 3.

3.6.3.1 Compiling matrix elements

Python has a variety of tools⁹ to evaluate mathematical expressions, however all that were tested were slower than what was desired.

A useful study which benchmarks math parsers was found, <https://github.com/ArashPartow/math-parser-benchmark-project>. In this study a comparison of the number of executions per second vs. the length of the arithmetic expression is done for a variety of POEM math parsers (Parse Once Evaluate Many times). The code was modified in order to account for the largest matrix element of a 3276×3276 size matrix using the excited state methodology; and each parser evaluated this expression 100,000 times with results shown in Table 3.9.

Table 3.9 C++ math parser comparison. Time (ns) represents the time required to evaluate the expression once. $\sum N$ evaluations represents the total sum of adding the numerical result of all 100,000 evaluations; which is done to check for numerical stability of the math parser. DNQ shows the math parsers that Did Not Qualify for the stated reason.

Rank	Math Parser	Time (ns)	$\sum N$ evaluations
1	ExprTk	28456.160	3175121.430 360 799 189 656 973
2	muparser 2.2.4	45474.680	3175121.430 360 799 189 656 973
3	MathExpr	259754.400	3175121.430 360 799 189 656 973
4	muparserx	294784.990	3175121.430 360 799 189 656 973
5	Lepton	306093.130	3175121.430 360 799 189 656 973
DNQ	atmsp 1.0.4	-	Out of memory
DNQ	ExprTkFloat	-	Incorrect result

⁹Examples include: eval, asteval, NumExpr, pandaseval, PyParsing, safeeval.

Table 3.9 shows that ExprTk^[76] outperforms the competition for large byte math expressions. This library is implemented in C++, however with the benefit of Python’s ability to call other languages, a wrapper can be used to implement this library in Python. Rather than write one from scratch, a wrapper has already been developed called `cexprtk`, written by Michael Rushton^[103] and is released under the Common Public License Version 1.0 (See Appendix B).

`cexprtk` works by generating a `exprtk::symbol_table` with preliminary input values for all the variables which is instantiated¹⁰ and the variables are added to it. Then an `exprtk::expression` is instantiated and the symbol table is registered with the expression instance. Finally a `exprtk::parser` is instantiated where both the expression object and the string form of the expression are passed to a method of the parser called `compile`. It is this compilation process which makes the library incredibly efficient at parsing mathematical expressions. The compilation of all the arithmetic matrix elements only needs to occur once at the beginning of the optimisation calculation. The symbol table is then updated with the new NLP values from the next optimisation step and the serialised expressions evaluated. This offers a substantial performance increase over inbuilt Python methods; with the downside of being very memory intensive. The serialised expressions require more RAM than the original byte size of the expressions; and on the available hardware limits the size of matrix which can be used. Another means of processing the matrices is to evaluate the arithmetic impromptu.

3.6.3.2 Evaluating matrix elements

In Section 3.6.3.1 `eval` was found to be much slower than the compiled `cexprtk` approach; however `eval` was the fastest solution out of the non-compiled methods. There are inherent security issues with using `eval`; as a user can use this as an option to run any code on the computer. If `eval(input())` is in the code and `os` library has been imported, someone could type into `input()`, `os.system('rm -R *')` which would delete all files in the users home directory. Due to the nature and context of the code being developed this should be a non-issue as the code is used only within the research group with trusted members, but is mentioned here for completeness.

Whilst evaluating the matrix elements from file is slower than building the AST and compiling; it is much less resource intensive as no intermediary results are stored

¹⁰To create an instance of an object in an OO programming language.

into memory. This method is much simpler than `cexprtk`, with a minimal example being

```

Python: eval example
1 # Setup dictionary of values
2 dict = {"A":A, "B":B, "C":C, "m1":m1, "m2":m2, "m3":m3,
    ↪    "z1":z1, "z2":z2, "z3":z3}
3
4 # Place the eval command into a function using list
    ↪ comprehension
5 def process_line(line):
6     return [0 if line == '0\n' else eval(line,dict)]
7 # Use numpy broadcasting
8 idx = (np.arange(mat_size)[: ,None]*File_mat +
    ↪    np.arange(mat_size)).ravel()
9 HHout = [linesHH[i] for i in idx]

```

3.6.3.3 Generating less general matrices

The lower triangular byte sizes seen in Table 3.8 for the excited state matrices can be reduced significantly by assigning known values for certain variables whilst the matrices are being generated in Maple. For example, a matrix can be generated for atomic systems where m_1, m_2 are electrons by setting the following; $m_1 = m_2 = 1, Z_1 = Z_2 = -1$ leaving just A, B, C, m_3 and Z_3 undefined, which makes it valid for the helium isoelectronic sequence and arbitrary nuclear charge/mass with three NLPs; however no molecular systems can be modelled using this matrix. This custom matrix creation loses the generality of the excited state matrices but makes the problem a lot more computationally tractable. Table 3.10 shows the byte sizes for the fully algebraic Hamiltonian matrix, \mathbf{H} compared to the same matrix generated for atoms where only m_3, Z_3, A, B are left undefined.

Table 3.10 Comparison of byte size for the general excited state, lower triangular Hamiltonian matrix, \mathbf{H} and the same matrix created for atoms in the excited state method for various values of ω . For the atomic matrix $m_1 = m_2 = 1, Z_1 = Z_2 = -1$ whilst m_3, Z_3, A, B, C are left in their general form. These sizes are compared against the fully general matrices where all variables are left undefined in their algebraic form.

\mathbf{H} , matrix byte size (Mega Bytes)		
ω	Fully general	Atoms
5	2.62	0.56
10	68.72	13.26
15	595.92	132.7
20	3017.13	689.53
25	11126.44	2590.17
30	24478.10	8915.04

3.6.3.4 Combination of cexprtk, eval and custom matrix creation

Using a combination of the above methods results in a routine which strikes a balance between computational cost and code versatility/accuracy of results. The following steps make up an optimised routine for calculation of the excited state matrices and solving of the generalised eigenvalue problem.

1. Maple reads the A, B, C recursion relation from file, generates and simplifies the kinetic energy matrix, \mathbf{T} , potential energy matrix, \mathbf{V} and overlap matrix, \mathbf{S} . All parameters can be left undefined, or values can be substituted for known types of system to make the calculation more computationally tractable, albeit losing the generality of the matrices. The algebraic \mathbf{T}, \mathbf{V} and \mathbf{S} matrices are saved to separate files.
2. The Python code accesses the matrix files without explicitly loading them into memory. The evaluation method of the arithmetic expressions in each of the matrices can be specified by the user, allowing for a combination of `cexprtk` and `eval` to be applied on different matrices. For small matrices `cexprtk` can be used on all matrices. For larger matrices (3000+) it is recommended that `cexprtk` be used on \mathbf{T} and \mathbf{V} as these are both larger than the overlap, \mathbf{S} , where `eval` can be used.
3. Once the arithmetic expressions have all been evaluated the matrix is then passed to the eigenvalue solver stage, where the method of diagonalisation can be specified as the `DSYEVD` or `QZ` method.

3.6.3.5 Using computer architecture - Multithreading

A computer's CPU consists of a number of physical cores and logical cores. Physical cores are the actual physical cores of the chipset, whereas logical cores are the abilities of a single core to do 2 or more things simultaneously. As an example, the Intel Core i5-2520M dual core CPU of the ThinkPad T420 laptop has two physical cores but four logical cores as each of the physical cores has two threads.

Multithreading is the ability of a computer's CPU to execute multiple processes or threads concurrently. On the surface it appears as though Python is not able to utilise features of multithreading due to the Global Interpreter Lock (GIL)^[104]. This protects access to Python objects, preventing multiple threads from executing Python bytecodes simultaneously. This means that native Python code is unable to be run in parallel.

The Python library `multiprocessing` side-steps the GIL by using subprocesses instead of threads, where each thread is given its own Python interpreter and GIL. The name of the module is somewhat confusing as `multiprocessing` usually refers to using two or more CPUs within a single computer system^[105]; whereas here it is referring to Python subprocesses. Care must be taken when spawning child processes due to a possibility of data corruption between threads. A very simple minimal working example is now presented on how child processes are spawned in order to evaluate a string of maths contained in a file using `cexprtk`. This code subdivides the file into 4 child processes which are each evaluated on a separate logical core of the CPU.

Python: Example of spawning child processes

```

1 # -*- coding: utf-8 -*-
2
3 import cexprtk
4 import timeit
5
6 # Setup the symbol table
7 st = cexprtk.Symbol_Table({"A":A, "B":B, "C":C, "m3":m3,
8   ↪  "z3":z3}, add_constants = True)
9
10
11 from multiprocessing import Pool
12
13 #Line is the line within the file

```

```

12 def process_line(line):
13     return cexprtk.Expression(line, st)
14
15 if __name__ == "__main__":
16     pool = Pool(4)
17     # chunk the work into 4 batches
18     start_time = timeit.default_timer()
19     results = pool.map(process_line, source_file, M/4)
20     print(timeit.default_timer() - start_time)

```

The principles of multithreading are used when evaluating the arithmetic within the matrices during each optimisation step.

If the user specifies `cexprtk` it will firstly compile all of the non-zero expressions¹¹ within the matrix into very efficient run time code. This process is only needed once at the start of the calculation and holds these serialised expressions into memory. These serialised expressions are then subdivided into j blocks, where j represents the number of logical cores the CPU has or how many the user requests to be used. If the user specifies `eval` then it does not need this initial compilation step and immediately sub-divides the matrix into the desired number of child processes. This process has been fully automated by having the Python code work out how many logical cores it can access on the computer and then subdivides the matrix accordingly, but can also be specified by the user in the run file. If the dimensions of the matrix are not divisible by the number of logical cores, it will create one block which carries the extra rows (e.g. a 95×95 matrix using 4 logical cores will be subdivided into 3 blocks of 23 rows and 1 block of 26 rows, $(3 \times 23) + 26 = 95$). Figure 3.5 shows a visual representation of this process.

¹¹N.B. Compiling just the non-zero matrix elements offers a substantial speed increase when compared to compiling all matrix elements, as compilation of all the zero's in the upper triangular matrix is computationally costly.

**If cexprt: All non-zero arithmetic matrix elements
compiled into fast, efficient run-time code**

If eval: No compilation

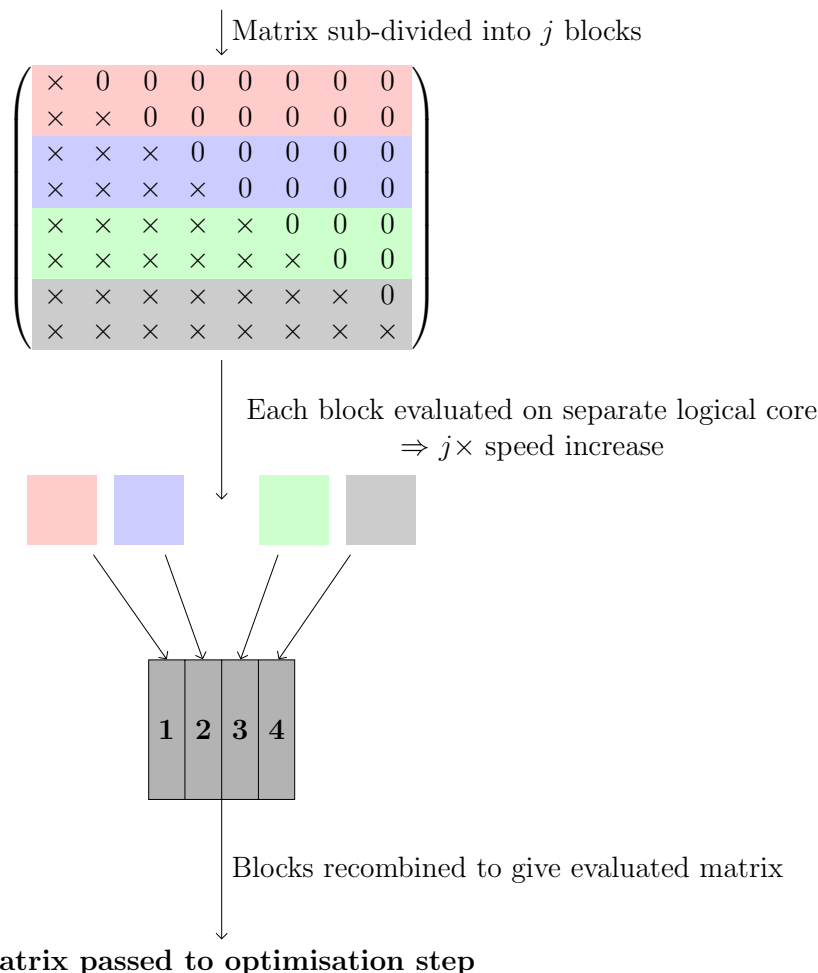


Figure 3.5 Steps involved in evaluating the arithmetic matrix elements making use of the CPU's logical cores. In this example a 8×8 lower triangular matrix is used with 4 logical cores, where \times 's represent algebraic non-zero matrix elements.

3.6.4 Calculation options

The excited state run file can be found in Appendix A where most options are self explanatory or have an accompanying description. As with the 3Body code, there are three main calculation options: OPT (optimisation), SP (Single Point) and SCAN (Scan parameter surface).

To run an optimisation calculation `Calc = 'OPT'` is set in the Python run file. The excited state Python code is able to vary up to three (A, B, C) NLPs; and various algorithms have been implemented either from SciPy's wrappers to C, or from custom written wrappers to C to minimise eigenvalues including:

-
- **Nelder-Mead**: Downhill simplex algorithm^[106].
 - **BOBYQA**: Bound optimisation by quadratic approximation^[91]. This was implemented with use of the NLOpt Python interface^[107].
 - **Powell**: Powell’s conjugate direction method^[108].
 - **Conjugate Gradient (CG)**: Python implementation of the CG algorithm^[82,83].
 - **Broyden-Fletcher-Goldfarb-Shanno (BFGS)**: Quasi-Newton method, hill-climbing optimisation^[92].
 - **L-BFGS-B**: Limited memory BFGS algorithm. This along with BFGS are best when a large number of variational parameters need optimising^[109].
 - **TNC**: Truncated Newton Algorithm^[110].
 - **COBYLA**: Constrained Optimisation BY Linear Approximation^[111].
 - **SLSQP**: Sequential Least Squares Programming^[112].
 - **dogleg**: Modified Newton method where the step is determined by the Powell method^[113].
 - **trust-ncg**: Newton conjugate gradient trust-region algorithm. Very similar to dogleg^[114].

Details of all these algorithms will not be provided here as detailed explanations can be found in their accompanying references.

There are two types of optimisation used within the excited state code, bounded and unbounded. Bounded optimisation lets the user constrain the NLPs to be within specified limits whereas unbounded allows the NLPs to take any value. Bounded optimisation is useful for PE surfaces with an awkward shape (e.g. singularities) or for problems where the location of the minimum is roughly known. In Section 3.4.1 the BOBYQA algorithm uses bounded optimisation as it is a trust-region algorithm. All the implemented algorithms mentioned above are capable of unbounded optimisation, but only BOBYQA, L-BFGS-B, SLSQP, TNC and COBYLA are capable of bounded optimisation.

The default algorithm for unbounded optimisation in the excited state code is the Nelder-Mead algorithm, whilst the default for bounded is SLSQP as both were found to perform best for their optimisation types. BOBYQA is a recent addition to the excited state code, so future work will involve a thorough test against SLSQP to determine which is better.

3.7 Bash

The Bourne Again SHell (BASH) was used to write a variety of scripts which are designed to reduce complexity, lessen errors and offer a better means of saving program run information and data. The shell scripts act as bridges between the various C++, Maple and Python codes; allowing efficient I/O between languages and the ability to control and run the more complicated programs with a simple interface.

3.7.1 Main Shell Scripts

Run Script

The run script is the main script that invokes the 3Body program. The user is able to specify masses, charges, wavefunction type, matrix size etc... from within this file. It also contains default presets for various quantities such as target tolerance of the program or NLP starting values, and offers some rudimentary checks to make sure a calculation has been setup correctly such as making sure Z_1 and Z_2 have the same sign with Z_3 having the opposite sign. The run file can be seen in Appendix A.

AllMethods

The AllMethods shell script builds upon the run file by allowing for nearly any calculation to be run from a single command line argument by passing the user arguments into the run script automatically. The syntax is as follows

Bash: AllMethods

```
1 AllMethods m1:z1 m2:z2 m3:z3 Mat_size psi_type
```

Where m_i, z_i refer to the masses and charges of the three particles. Particle symbols, e, p, d, t, He etc... are predefined in a separate file which inputs the latest CODATA values; or arbitrary values can be specified. Mat_size is the required matrix size and psi_type is the wavefunction type to be used with options being, PEKERIS, K, A, AC, ABC, with the excited state code handled separately due to more unique options. There are various optional arguments that AllMethods recognises to setup and run a wide variety of calculation types:

- -p: Prepare a run file, but do not execute it.
- -c: Run a critical charge calculation.
- -f: Run a fork calculation (NROpt and BOBYQA simultaneously).

- `-v`: After calculation of system, recalculate the eigenvector with 32 digits of precision (single point).
- `-vo`: After calculation of system, recalculate the eigenvector with 32 digits of precision (optimisation).
- `-nu`: After calculation, calculate the cusps.
- `-ia`: After calculation, calculate the inner and outer expectation values.
- `-id`: After calculation, calculate the inner and outer particle densities (RDF).
- `-as`: Calculate using the antisymmetric wavefunction.
- `-rX`: RootX Calculates the Xth eigenvalue and eigenvector (1 is the first root, the ground state). The default is `-r1`
- `?`: Prints the possible options to the terminal.

As an example of using these optional arguments, consider running a calculation for the ground state of fixed nucleus helium using a 2856 size matrix and AC wavefunction, followed by calculating the inter-particle cusps.

Bash: Example of AllMethods optional arguments

```
1 AllMethods -nu e e infinity:2 2856 AC
```

CalcExpectation

This script calculates various user specified expectation values by passing the input to the Expectation value code outlined in Section 3.4.2.

Bash: CalcExpectation syntax

```
1 CalcExpectation ExpValName SVFILE|LOGFILE sym|antisym|asym
  ↪ PEKERIS|K|A|AC|ABC [-arb <NUM>]
```

SVFILE is the text file that holds the wavefunction, LOGFILE is the text file where the output of the C++ program was piped to. If the sv file is used, the expectation value calculation is only printed to terminal whereas using the log file will append the output to the end of the log file. ExpValName is the name of the expectation value, and “|” represents “or” with one of the set of arguments being required. The possible expectation value names are:

- **ri**: Expectation value of r_i , $\langle r_i \rangle$, $i = 1, 2, 3$
- **ri^2**: Expectation value of r_i^2 , $\langle r_i^2 \rangle$, $i = 1, 2, 3$
- **ri^3**: Expectation value of r_i^3 , $\langle r_i^3 \rangle$, $i = 1, 2, 3$
- **ri^-1**: Expectation value of r_i^{-1} , $\langle r_i^{-1} \rangle$, $i = 1, 2, 3$
- **dri**: Expectation value of $\langle \delta(r_i) \rangle$, $i = 1, 2, 3$
- **taui**: Expectation value of inter-particle cosine function, τ_i , $\langle \tau_i \rangle$, $i = 1, 2, 3$
- **T**: Expectation value of \hat{T} , $\langle \hat{T} \rangle$
- **V**: Expectation value of \hat{V} , $\langle \hat{V} \rangle$

If all the expectation values are required, then the `CalcExpectations` bash script can be used instead.

Bash: CalcExpectations

```
1 CalcExpectations SVFILE|LOGFILE sym|antisym|asym
  ↪ PEKERIS|K|A|AC|ABC [-arb <NUM>]
```

AllMethodsEX

The excited state code has unique user options so a separate bash script was written to control this problem called, `AllMethodsEX`. The syntax for this script is as follows

Bash: AllMethodsEX

```
1 AllMethodsEX m3 Z3 MATSIZE OPT|SP|SCAN A|AC|AB|ABC ROOT PRS
  ↪ SOLVER NEARESTEIG A B C
```

Many of these options are the same as in the `AllMethods` script, with exceptions being:

- **PRS**: The number of processors used to evaluate the excited state matrices if using the `eval` method.
- **SOLVER**: The eigensolver to use, either `syevd` or `QZ`.
- **NEARESTEIG**: If an eigenvalue is known, then it can be input here to aid the `QZ` algorithm.
- **A B C**: Input the starting A , B and C .

CalcCusps

This code takes a log file/sv file and looks through the log file (if given) to determine if the denominator of equation (2.42) has already been calculated and appended by `CalcExpectation(s)`. If it has, the Maple code is given the value and if not, the Maple code will explicitly calculate it which is slower.

Bash: CalcCusps syntax

```
1 CalcCusps LOGFILE/SVFILE
```

ReCalcVector

The purpose of this useful bash script is to take the output from an optimisation conducted at double precision and recalculate the eigenvector as a single point with higher precision (`-arb`)

Bash: ReCalcVector

```
1 ReCalcVector --arb 32 LOGFILE
```

This script can also be used to set up and run an optimisation calculation with the starting point as the optimised NLPs from the log file, but at higher precision. This can be done using the `-opt` command-line option.

3.8 Summary and Conclusions

An overview of the C++, Maple and Python programs have been given which are used to calculate all the results presented in this thesis. Some of the key Bash scripts are also discussed which are used to bridge together all the larger codes. The C++ `3Body` program calculates the eigenvalues and eigenvectors for the ground state problem. Expectation values are calculated using `Expectation` and the CUBA library is used to calculate particle densities and certain expectation values. Maple is used to calculate and save the 57-term recursion relation for use by `3Body`, and the excited state matrices which are used by the main Python code to calculate eigenvalues and eigenvectors of excited state atoms and ion. The Bash scripts are designed to automate, lessen user input error and keep program records. Licenses for the libraries used in this work can be found in Appendix B. Throughout this chapter, possible program improvements have been suggested, including alternative algorithms, infrastructure and libraries. These provide direction for future improvements and increasing the efficiency of the variety of codes written and used in this work.

CHAPTER 4



Nuclear motion in three-particle diatomic ions

4.1 Introduction

The Born Oppenheimer approximation (BO) has long reigned as one of the most important and fundamental approximations in molecular quantum mechanics^[115,116], with details of this approximation given in Chapter 2 Section 2.8. In most instances, it is a very effective tool in providing insight into molecular structure and is capable of accurate results despite being a major approximation. There are instances where it falters such as being unable to account for the occurrence of conical intersections where the coupling between electronic and nuclear motion becomes important^[117]. With the constant advances in theoretical and experimental methods^[118–123] along with computational technology, it is now possible to probe the motion of nuclei and to understand the importance of nuclear motion on the fundamental physics of particle interaction. Experimental physics is consistently becoming more sophisticated and can now measure time at the atto-second timescale, meaning the effects of nuclear and electronic motion can be measured with greater accuracy. As these experiments improve it is vital that theory remains competitive in order to explain the underlying chemical physics.

Here, the motion of heavy nuclei a_i^+ , $i = 1, 2$ in three-particle molecular systems $\{a_1^+, a_2^+, a_3^-\}$ is studied with no *a-priori* approximations to ensure no bias or assumptions are introduced in the system. The particles interact *via* a Coulomb potential with finite particle masses used at all times, hence there is no reliance on the Born Oppenheimer approximation. The wavefunctions and energies for all systems studied were calculated using the series solution method described in Chapter 2.

The Intracule density is a pair density which measures the radial correlation between the two nuclei, a_1^+ and a_2^+ , and provide information on the dynamical behaviour of the two particles and their equilibrium distance. This means they act as a means of studying nuclear motion in more detail, and how increasing the mass of the heavy particles a_1^+ and a_2^+ effects this equilibrium distance. Arias de Saavedra *et al.*^[19] reported intracule densities for electronic and muonic three-particle molecular

systems using the same Laguerre based wavefunction used in this work. A pair density offers some insight into the effects of nuclear motion, but it is also beneficial to understand how each particle moves separately with respect to other particles in the system. Mátyus *et al.*^[62] calculated the particle density at the centre of mass to study the transition from atomic $\{e^-e^-p^+\}$ to molecular $\{p^+p^+e^-\}$ systems using explicitly correlated Gaussian functions and translationally invariant Cartesian coordinates. This is one of the main starting points for the work conducted in this chapter. By measuring particle densities from a relative point (the centre of mass) it has been found that it is possible to uniquely define the correlated motion of the heavy masses a_1^+ and a_2^+ individually rather than look at their combined relative effects provided by intracule densities.

Chapter summary

The aim of this chapter is to go beyond the Born Oppenheimer approximation and investigate the effects of nuclear motion in three particle homonuclear and heteronuclear diatomic ions. This is done with the use of intracule and centre of mass particle densities, of which the latter reveals a nuclear correlation not attainable when assuming the Born Oppenheimer approximation. Research presented in this chapter has been published in Physical Review A^[23]. Some figures and tables have been reproduced from this publication, © 2016 American Physical Society.

4.2 Method

4.2.1 Barycentric coordinates

The wavefunction calculated from solving the Schrödinger equation in this work is represented in perimetric coordinates, z_i , which are a linear combination of the interparticle distances r_i such that $z_i = r_j + r_k - r_i$. Using the method described in Chapter 2 the centre of mass motion is separated off and a translation free problem is considered, due to the continuous spectrum of translational motion not being quantised in quantum mechanics. This means that in order to find the centre of mass distance from one of the heavy particles, a coordinate transformation is required from inter-particle coordinates to one which has this distance explicitly defined. Various coordinate systems were investigated including Cartesian coordinates, trilinear coordinates and Radau vectors, but a more natural choice for this coordinate system is barycentric coordinates, as the barycentre of a triangle is defined as the centre of mass of a triangle^[124].

Barycentric coordinates can also be found by the name “area” coordinates^[124]. This name highlights the underlying premise of barycentric coordinates in that their value signifies the ratio of the areas of each sub-triangle formed by a point within the triangle, and the total area of the triangle.

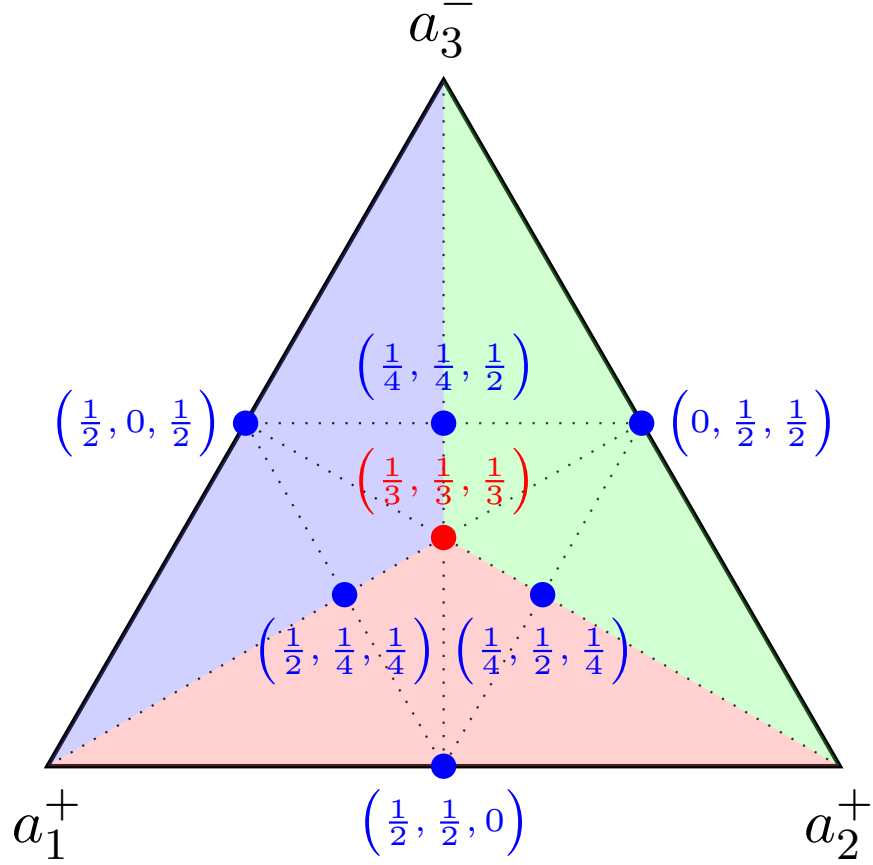


Figure 4.1 Representation of barycentric coordinates $(\lambda_1, \lambda_2, \lambda_3)$. When all masses are equal, i.e. $m_{a_1^+} = m_{a_2^+} = m_{a_3^-}$ this corresponds to an equilateral triangle with the masses located at the vertices. The red dot corresponds to the barycentre which is the centre of the triangle and other various barycentric coordinates have been labelled in blue. The three sub-triangles formed around the barycentre have been highlighted in blue, red and green which represent the triangular sub regions created by the centre of mass in the ratio $\frac{1}{3}, \frac{1}{3}, \frac{1}{3}$.

Figure 4.1 shows the barycentric coordinate system, various points and their respective coordinates within the triangle. The vertices of the triangle are given by $(1,0,0)$, $(0,1,0)$ and $(0,0,1)$ and every point within this triangle can be written as a unique convex combination¹ of the three vertices^[124]. These coordinates are given by $\lambda_i = m_{a_i}/(m_{a_i} + m_{a_j} + m_{a_k})$ and there is a unique sequence of three numbers,

¹A mapping of pairs of points with pairs of non-negative real numbers, to another point of the space^[125].

$\lambda_1, \lambda_2, \lambda_3 \geq 0$ such that $\lambda_1 + \lambda_2 + \lambda_3 = 1$. Note that although there are three coordinates, there are only two degrees of freedom, since $\lambda_1 + \lambda_2 + \lambda_3 = 1$; thus every point is uniquely defined by any two of the barycentric coordinates.

Barycentric coordinates are incredibly useful in multiple branches of mathematics, computer graphics and engineering^[124,126,127], but they lack a key feature which is required within this work. The coordinates themselves are ratios of sub-triangle areas, however to calculate a particle density a physical distance is required. In order to do this a new coordinate system was developed which uses a combination of inter-particle coordinates and barycentric coordinates, thus explicitly defining the centre of mass distance.

4.2.2 New Coordinate System

The inter-particle coordinate system can be seen in Figure 4.2 (a). The three inter-particle coordinates are not a suitable coordinate system with which to find the centre of mass of a three particle system as they only connect the vertices of the triangle. The three distances specify the domain of the triangle and all the points within, but they are unable to extract out the distance from any one of the three particles to the centre of mass position in a computationally effective manner. The distance from one of the heavy masses a_1^+ or a_2^+ to the centre of mass, c , is required and have been labelled $s_i, i = 1$ or 2 which can be seen in Figure 4.2 (b).

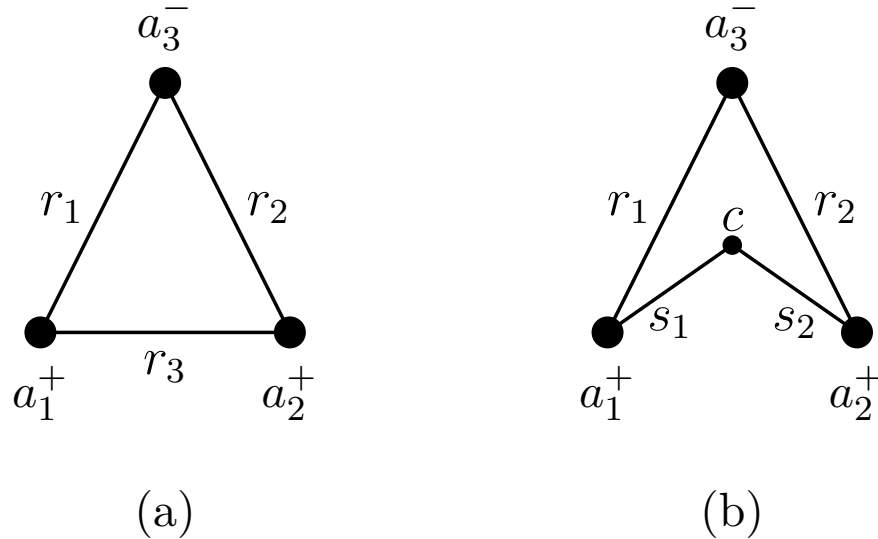


Figure 4.2 Coordinate systems for $\{a_1^+ a_2^+ a_3^-\}$ systems: (a) inter-particle coordinates r_1, r_2 and r_3 , and (b) centre of mass coordinates r_1, r_2 and $s_i, i = 1$ or 2 , where c is the centre of mass.

The wavefunction is expressed using three perimetric coordinates which are linear combinations of the three inter-particle coordinates r_1, r_2, r_3 . It is preferential to keep three coordinates so that in later integration steps, no more than three integrals are required which will speed up computation time.

Consider the situation shown in Figure 4.3 for a generic three-particle system. No *a priori* assumptions are made about the origin of the coordinate system, and hence the discussion is treated generally.

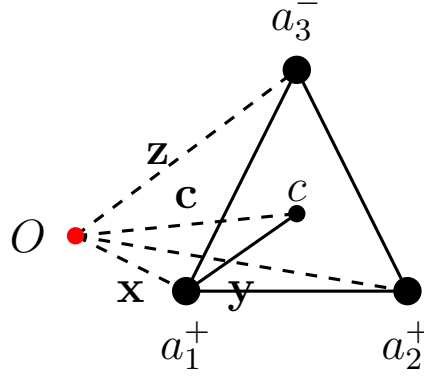


Figure 4.3 A generalised triangle of three particles $\{a_1^+, a_2^+, a_3^-\}$ relative to an arbitrary origin O (red dot).

Let \mathbf{x} be the position vector of a_1^+ , \mathbf{y} the position vector of a_2^+ and \mathbf{z} the position vector of a_3^- with respect to the origin O . The centre of mass point c with normalised barycentric coordinates $(\lambda_1, \lambda_2, \lambda_3)$ where $\lambda_1 + \lambda_2 + \lambda_3 = 1$, will have a position vector given by

$$\mathbf{c} = \lambda_1 \mathbf{x} + \lambda_2 \mathbf{y} + \lambda_3 \mathbf{z}. \quad (4.1)$$

This can be simplified by moving the origin to one of the heavy particles say, a_1^+ which then does not require the inclusion of the mass a_1^+ and the mass ratio λ_1

$$\mathbf{c} - \mathbf{x} = \lambda_1(\mathbf{x} - \mathbf{x}) + \lambda_2(\mathbf{y} - \mathbf{x}) + \lambda_3(\mathbf{z} - \mathbf{x}). \quad (4.2)$$

Simplifying gives

$$\mathbf{c} - \mathbf{x} = \lambda_2(\mathbf{y} - \mathbf{x}) + \lambda_3(\mathbf{z} - \mathbf{x}). \quad (4.3)$$

Representing this form in terms of inter-particle coordinates

$$\mathbf{s}_1 = \lambda_2 \mathbf{r}_3 + \lambda_3 \mathbf{r}_1. \quad (4.4)$$

The scalar distance, s_1 is desired and the following dot product is calculated

$$|\mathbf{s}_1|^2 = \mathbf{s}_1 \cdot \mathbf{s}_1 = (\lambda_2 \mathbf{r}_3 + \lambda_3 \mathbf{r}_1) \cdot (\lambda_2 \mathbf{r}_3 + \lambda_3 \mathbf{r}_1). \quad (4.5)$$

Expanding the brackets

$$|\mathbf{s}_1|^2 = \lambda_2^2 |\mathbf{r}_3|^2 + \lambda_3^2 |\mathbf{r}_1|^2 + 2\lambda_2 \lambda_3 \mathbf{r}_3 \cdot \mathbf{r}_1. \quad (4.6)$$

Upon application of the dot product to each of the terms

$$|\mathbf{s}_1|^2 = \lambda_2^2 r_3^2 + \lambda_3^2 r_1^2 + 2\lambda_2 \lambda_3 r_3 r_1 \cos \alpha, \quad (4.7)$$

where $|\mathbf{s}_1| = s_1$ is the length of \mathbf{s}_1 , and α is the angle $\angle(a_2^+, a_1^+, a_3^-)$. Using the cosine rule, the cosine term can be removed by rearranging the following form of r_2 in terms of $\cos \alpha$ and substituting into the form for $|\mathbf{s}_1|^2$ given in equation (4.7), i.e.,

$$\begin{aligned} r_2^2 &= r_1^2 + r_3^2 - 2r_1 r_3 \cos \alpha \\ \therefore \cos \alpha &= -\frac{r_2^2 - r_1^2 - r_3^2}{2r_1 r_3}. \end{aligned}$$

The distance $|\mathbf{s}_1|$ now satisfies:

$$s_1^2 = (r_1 \lambda_3)^2 + (r_3 \lambda_2)^2 + \lambda_2 \lambda_3 (r_1^2 + r_3^2 - r_2^2). \quad (4.8)$$

Equation (4.8) gives the expression for s_1 , the distance from the heavy particle a_1^+ to the centre of mass position c in terms of the inter-particle distances. This is an undesirable form as the distance s_1 depends on all three of the inter-particle distances, r_i , which will create a bottleneck in calculations due to the four integrals required in order to correctly map the space. To overcome this, the dependency of the integration on r_3 can be removed. Rearranging s_1 in equation (4.8) in terms of r_3

$$r_3 = \pm \frac{\sqrt{(-\lambda_2(\lambda_2 + \lambda_3))(\lambda_2 \lambda_3 r_1^2 - \lambda_2 \lambda_3 r_2^2 + \lambda_3^2 r_1^2 - s_1^2)}}{\lambda_2(\lambda_2 + \lambda_3)}. \quad (4.9)$$

Within the r_1, r_2, r_3 coordinate system the integration domains for two of the inter-particle coordinates are $0 \rightarrow \infty$ whereas the integration domain for the third depends on the first two coordinates as a consequence of the triangular condition^[12]. Making the domain of r_1 depend on r_2 and r_3 and substituting the positive form of r_3 from equation (4.9) into these ranges $|r_2 - r_3| \rightarrow r_2 + r_3$ and then solving each one for r_1 results in the new integration domains for r_1 . This process is shown below.

The lower limit for the integration domain for r_1 is $|r_2 - r_3|$, and by substituting equation (4.9) into this domain results in

$$r_1 = |r_2 - r_3| = \left| r_2 - \frac{\sqrt{-\lambda_2(\lambda_2 + \lambda_3)(\lambda_2\lambda_3r_1^2 - \lambda_2\lambda_3r_2^2 + \lambda_3^2r_1^2 - s_1^2)}}{\lambda_2(\lambda_2 + \lambda_3)} \right| \quad (4.10)$$

The upper limit for the integration domain for r_1 is $r_2 + r_3$, and by substituting equation (4.9) into this domain results in

$$r_1 = r_2 + r_3 = r_2 + \frac{\sqrt{-\lambda_2(\lambda_2 + \lambda_3)(\lambda_2\lambda_3r_1^2 - \lambda_2\lambda_3r_2^2 + \lambda_3^2r_1^2 - s_1^2)}}{\lambda_2(\lambda_2 + \lambda_3)}. \quad (4.11)$$

Solving equations (4.10) and (4.11) for r_1 results in the following integration domains for the centre of mass particle density

$$\begin{aligned} r_1 : \left| \frac{r_2\lambda_2 - s_1}{\lambda_2 + \lambda_3} \right| &\rightarrow \frac{r_2\lambda_2 + s_1}{\lambda_2 + \lambda_3} \\ r_2 : 0 &\rightarrow \infty \\ s_1 : 0 &\rightarrow \infty \end{aligned} \quad (4.12)$$

The new coordinate system can be expressed in terms of r_1, r_2, s_1 , for the instance of measuring the particle density from the centre of mass to particle a_1^+ , and appropriate permutation of coordinates will allow the other centre of mass distances s_2 and s_3 to be found.

The integration domains derived above to calculate the particle density from the heavy particle a_1^+ can be combined with the Dirac delta function to give the following integration form

$$\rho_{c,a_1}(s) = \underbrace{\int_0^\infty}_{s_1} \underbrace{\int_0^\infty}_{r_2} \underbrace{\int_{\left| \frac{r_2\lambda_2 - s_1}{\lambda_2 + \lambda_3} \right|}^{\frac{r_2\lambda_2 + s_1}{\lambda_2 + \lambda_3}}}_{r_1} \Psi(r_1, r_2, s_1) \delta(s_1 - s) \Psi(r_1, r_2, s_1) dV. \quad (4.13)$$

To find the volume element dV the Jacobian factor needs to be considered to account for a 1 : 1 mapping between coordinate spaces going from the inter-particle coordinate system and the new centre of mass coordinates. The Jacobian, J , is given by

$$J = \left| \frac{\partial(r_1, r_2, r_{12})}{\partial(r_1, r_2, s_1)} \right| = \left| \begin{array}{ccc} \frac{\partial r_1}{\partial r_1} & \frac{\partial r_1}{\partial r_2} & \frac{\partial r_1}{\partial s_1} \\ \frac{\partial r_2}{\partial r_1} & \frac{\partial r_2}{\partial r_2} & \frac{\partial r_2}{\partial s_1} \\ \frac{\partial r_3}{\partial r_1} & \frac{\partial r_3}{\partial r_2} & \frac{\partial r_3}{\partial s_1} \end{array} \right| = \frac{s_1}{\sqrt{-\lambda_2(\lambda_2 + \lambda_3)(\lambda_2\lambda_3r_1^2 - \lambda_2\lambda_3r_2^2 + \lambda_3^2r_1^2 - s_1^2)}} \quad (4.14)$$

The volume element in terms of r_1 , r_2 and r_3 is $dV = 8\pi^2 r_1 r_2 r_3 dr_1 dr_2 dr_3$ ^[12]. Looking at equation (4.14), the denominator is equal to the numerator of equation (4.9). Substituting the expression for r_3 into the inter-particle volume element and multiplying by the Jacobian given in equation (4.14), the volume element for the new coordinate system in terms of r_1 , r_2 and s_1 is given by

$$\begin{aligned} dV = 8\pi^2 r_1 r_2 \times & \left. \frac{\sqrt{-\lambda_2(\lambda_2 + \lambda_3)(\lambda_2\lambda_3r_1^2 - \lambda_2\lambda_3r_2^2 + \lambda_3^2r_1^2 - s_1^2)}}{\lambda_2(\lambda_2 + \lambda_3)} \right\} r_3 \\ & \times \left. \frac{s_1}{\sqrt{-\lambda_2(\lambda_2 + \lambda_3)(\lambda_2\lambda_3r_1^2 - \lambda_2\lambda_3r_2^2 + \lambda_3^2r_1^2 - s_1^2)}} \right\} J \\ & \times dr_1 dr_2 ds_1 \end{aligned} \quad (4.15)$$

Simplifying gives the volume element for the centre of mass particle densities

$$dV = \frac{8\pi^2 r_1 r_2 s_1}{\lambda_2(\lambda_2 + \lambda_3)} dr_1 dr_2 ds_1. \quad (4.16)$$

4.3 Implementation

The centre of mass particle density from the heavy particle a_1^+ (with the Jacobian, equation (4.16), omitted for simplicity) is

$$\rho_{c,a_1}(s) = \underbrace{\int_0^\infty}_{s_1} \underbrace{\int_0^\infty}_{r_2} \underbrace{\int_{\left| \frac{r_2\lambda_2 - s_1}{\lambda_2 + \lambda_3} \right|}^{\frac{r_2\lambda_2 + s_1}{\lambda_2 + \lambda_3}}}_{r_1} \Psi(r_1, r_2, s_1) \delta(s_1 - s) \Psi(r_1, r_2, s_1) dr_1 dr_2 ds_1. \quad (4.17)$$

Integrating over the Dirac delta function results in the following form

$$\rho_{c,a_1}(s) = \int_0^\infty \int_{\left| \frac{r_2\lambda_2 - s}{\lambda_2 + \lambda_3} \right|}^{\frac{r_2\lambda_2 + s}{\lambda_2 + \lambda_3}} \Psi(r_1, r_2, s) \Psi(r_1, r_2, s) dr_1 dr_2, \quad (4.18)$$

where λ_i are mass ratios, $\lambda_i = m_i/(m_i + m_j + m_k)$ and s is the distance from particle a_i (which in this example is a_1) to the centre of mass. The integration is very computationally inefficient in this form so methods were used to simplify the calculation. The goal was to make the inner integral limits numerical rather than algebraic, which can be done by performing a mapping on its integration domain. Due to the modulus sign in the lower limit it is not possible in its current form to do this, as unique numerical integration limits cannot be found. Instead the integration can be decomposed into the following

$$\rho_{c,a_1}(s) = \underbrace{\int_0^{\frac{s}{\lambda_2}} \int_{\frac{s-r_2\lambda_2}{\lambda_2+\lambda_3}}^{\frac{s+r_2\lambda_2}{\lambda_2+\lambda_3}} \Psi(r_1, r_2, s) \Psi(r_1, r_2, s) dr_1 dr_2}_{\mathbf{1}} + \underbrace{\int_{\frac{s}{\lambda_2}}^{\infty} \int_{\frac{r_2\lambda_2-s}{\lambda_2+\lambda_3}}^{\frac{r_2\lambda_2+s}{\lambda_2+\lambda_3}} \Psi(r_1, r_2, s) \Psi(r_1, r_2, s) dr_1 dr_2}_{\mathbf{2}}. \quad (4.19)$$

By doing this, the modulus sign has been removed from the integration domains and now it is possible to do a mapping to make the limits numerical. Introducing the new variable to integrate over ξ , via the following substitution

$$\xi = \frac{(\lambda_2 + \lambda_3)r_1 - s}{r_2\lambda_2}. \quad (4.20)$$

The original volume element dr_1 is then substituted with the following

$$\frac{d\xi}{dr_1} = \frac{\lambda_2 + \lambda_3}{r_2\lambda_2} \therefore dr_1 = \frac{r_2\lambda_2}{\lambda_2 + \lambda_3} d\xi. \quad (4.21)$$

The integration limits for integral **1** must be changed accordingly, as follows.

$$\int_{\underbrace{\frac{s-r_2\lambda_2}{\lambda_2+\lambda_3}}_1}^{\frac{s+r_2\lambda_2}{\lambda_2+\lambda_3}} \Psi(r_1, r_2, s) \Psi(r_1, r_2, s) dr_1$$

Lower limit	Upper limit
$r_1 = \frac{s-r_2\lambda_2}{\lambda_2+\lambda_3}$	$r_1 = \frac{s+r_2\lambda_2}{\lambda_2+\lambda_3}$
\downarrow	\downarrow
Substitute into equation (4.20):	Substitute into equation (4.20):
$\xi = \frac{(\lambda_2+\lambda_3)\left(\frac{s-r_2\lambda_2}{\lambda_2+\lambda_3}\right)-s}{r_2\lambda_2}$	$\xi = \frac{(\lambda_2+\lambda_3)\left(\frac{s+r_2\lambda_2}{\lambda_2+\lambda_3}\right)-s}{r_2\lambda_2}$
\downarrow	\downarrow
Cancelling terms gives:	Cancelling terms gives:
$\xi = -1$	$\xi = 1$

These are the new integration limits for integral one.

Including the volume element and Jacobian gives the following:

$$\int_{\underbrace{\frac{s-r_2\lambda_2}{\lambda_2+\lambda_3}}_1}^{\frac{s+r_2\lambda_2}{\lambda_2+\lambda_3}} \Psi(r_1, r_2, s) \Psi(r_1, r_2, s) dr_1 \equiv \int_{\underbrace{-1}_1}^1 \frac{\Psi(\xi, r_2, s) \Psi(\xi, r_2, s) r_2 \lambda_2}{\lambda_2 + \lambda_3} d\xi$$

Following these alterations, the first part of equation (4.19) now reads

$$\int_0^{\frac{s}{\lambda_2}} \int_{\underbrace{-1}_1}^1 \frac{\Psi(\xi, r_2, s) \Psi(\xi, r_2, s) r_2 \lambda_2}{\lambda_2 + \lambda_3} d\xi dr_2. \quad (4.22)$$

For integral **2** in equation (4.19) the new variable ζ is introduced, which is defined as

$$\zeta = \frac{r_1(\lambda_2 + \lambda_3) - r_2\lambda_2}{s}. \quad (4.23)$$

The existing volume element dr_1 is rewritten in terms of $d\zeta$

$$\frac{d\zeta}{dr_1} = \frac{\lambda_2 + \lambda_3}{s} \therefore dr_1 = \frac{s}{\lambda_2 + \lambda_3} d\zeta. \quad (4.24)$$

The limits of integration for integral **2** are modified *via* the following procedure.

$$\int_{\underbrace{\frac{r_2\lambda_2-s}{\lambda_2+\lambda_3}}_2}^{\frac{r_2\lambda_2+s}{\lambda_2+\lambda_3}} \Psi(r_1, r_2, s) \Psi(r_1, r_2, s) dr_1$$

Lower limit	Upper limit
$r_1 = \frac{r_2\lambda_2-s}{\lambda_2+\lambda_3}$	$r_1 = \frac{r_2\lambda_2+s}{\lambda_2+\lambda_3}$
↓	↓
Substitute into equation (4.23):	Substitute into equation (4.23):
$\zeta = \frac{\frac{r_2\lambda_2-s}{\lambda_2+\lambda_3}(\lambda_2+\lambda_3)-r_2\lambda_2}{s}$	$\zeta = \frac{\frac{r_2\lambda_2+s}{\lambda_2+\lambda_3}(\lambda_2+\lambda_3)-r_2\lambda_2}{s}$
↓	↓
Cancelling terms gives:	Cancelling terms gives:
$\zeta = -1$	$\zeta = 1$

These are the new integration limits for integral two.

Including the volume element and Jacobian gives the following:

$$\int_{\underbrace{\frac{r_2\lambda_2-s}{\lambda_2+\lambda_3}}_2}^{\frac{r_2\lambda_2+s}{\lambda_2+\lambda_3}} \Psi(r_1, r_2, s) \Psi(r_1, r_2, s) dr_1 \equiv \int_{\underbrace{-1}_2}^1 \frac{\Psi(\zeta, r_2, s) \Psi(\zeta, r_2, s) s}{\lambda_2 + \lambda_3} d\zeta$$

Following these alterations, the second part of equation (4.19) now reads

$$\int_{\frac{s}{\lambda_2}}^{\infty} \int_{\underbrace{-1}_2}^1 \frac{\Psi(\zeta, r_2, s) \Psi(\zeta, r_2, s) s}{\lambda_2 + \lambda_3} d\zeta dr_2. \quad (4.25)$$

Combining the two parts, the new form of the centre of mass particle density moving away from the heavy mass a_1^+ reads

$$\rho_{c,a_1}(s) = \int_0^{\frac{s}{\lambda_2}} \underbrace{\int_{-1}^1}_1 \frac{\Psi(\xi, r_2, s) \Psi(\xi, r_2, s) r_2 \lambda_2}{\lambda_2 + \lambda_3} d\xi dr_2 + \int_{\frac{s}{\lambda_2}}^{\infty} \int_{\underbrace{-1}_2}^1 \frac{\Psi(\zeta, r_2, s) \Psi(\zeta, r_2, s) s}{\lambda_2 + \lambda_3} d\zeta dr_2. \quad (4.26)$$

The integration ranges are now numerical rather than algebraic, greatly speeding up their computation. The calculation remains demanding but since it is now numerical, the CUBA library^[97] can be used to speed up the evaluations, discussed in Chapter

3. In particular, the Cuhre multidimensional integration algorithm is well suited to the form of the integrands. The evaluation of the original form of the centre of mass (COM) density, equation (4.18), took days in Maple (2856 size matrix, $p^+p^+e^-$). Conversely, the new form for the COM density can be evaluated in seconds by building the integrands in Maple and passing to the highly optimised C++ code.

Running the program

Like all the codes presented in this work, the centre of mass particle density calculation is executable from the command line, using the following command.

Centre of mass command-line syntax

```
1 Calcdri -pd com SV/LOGFILE
```

4.4 Results and Discussion

4.4.1 Comparison with Born-Oppenheimer approximation

When investigating the effects of nuclear motion in a system, comparison should normally be made to the fixed nucleus Born-Oppenheimer approximation scheme. It is not possible to calculate intracule densities or COM densities under the BO framework: as two particles are fixed without uncertainty in their respective positions, so the particle densities would yield a delta function. For an overview of the BO approximation consult Chapter 2 Section 2.8.

A major advantage of treating a molecule nonadiabatically is that the kinematic effects of rotation and vibration are automatically included in the solution and do not need to be considered separately as per the BO approximation. Table 4.1 provides the non-adiabatic (fully-correlated) ground state energies and the expectation value of the internuclear distances $\langle r_3 \rangle$ of the electronic and muonic hydrogen molecule isotopologues.

These are compared with the zero-point energy (ZPE) corrected Born-Oppenheimer (BO) energy, and bond-length obtained from a standard computational chemistry calculation using the Gaussian09 software package^[32] at the Hartree-Fock (HF) level of theory with a very large aug-cc-pV6Z basis set. HF theory is equivalent to Configuration Interaction (CI) or Coupled Cluster (CC) methods for one-electron systems at the minimum of the potential energy surface for a given basis set.

As the mass of the nuclei increase, the kinetic energy decreases, which in the limit of infinite nuclear mass in the BO approximation becomes zero. This is evident

in the ZPE corrections provided by a frequency calculation at the minimum of the BO potential energy surface (as $E_{BO} = -0.602\,632\,9$ a.u. for all isotopologues). This energy stabilisation as the nuclear masses increase is also seen for the muonic hydrogen molecule isotopologues which have been included in Table 4.1.

Experimental dissociation energies and equilibrium bond distances (where available) have also been included in Table 4.1 along with those calculated from the current work and the ZPE-corrected BO values. The inter-nuclear expectation values, $\langle r_3 \rangle$, corresponding to the bond length, r , are all in excellent agreement with those of Nakashima *et al.*^[30]. These authors have attributed the shortening of the bond length $\langle r_3 \rangle$ to the electrons being more attracted to the nucleus as its mass becomes heavier^[30], which can not be explained in the BO approximation.

The Fully Correlated (FC) dissociation energies, D_0 , were calculated as the difference between the full, three-body energy and the two-body energy, E_{ij} ; and the BO dissociation energies, $D_0(\text{BO})$ were calculated as the difference between the ZPE-corrected BO energies and the two-body energy, E_{ij} . Table 4.1 demonstrates that to obtain very accurate dissociation energies and explain variations in structural data of the isotopologues, it is important to include the coupling of the electronic and nuclear motion. The data for muonic systems are also in good agreement with the work of Bhatia and Drachman^[128]. The muonic systems show much higher dissociation energies than their electronic counterparts, explained by analysing the energy of the two-particle system resulting from dissociation which is always the heavier, oppositely charged pair. The energy of a two-particle system, E_{ij} is known analytically as^[53]

$$E_{ij} = -\frac{Z_i Z_j}{2n} \frac{1}{m_i^{-1} + m_j^{-1}}, \quad (4.27)$$

where $n = 1$ for the ground state. Consider: 1) H_2^+ , $(p^+ p^+ e^-)$ and 2) μH_2^+ , $(p^+ p^+ \mu^-)$. For 1) $E_{p^+ e^-} = -0.499\,727\,E_h$ whereas for 2) $E_{p^+ \mu^-} = -92.920\,417\,E_h$ demonstrating that a stronger attractive interaction is occurring due to the greater mass of the muon.

Furthermore, the energy difference between the energy eigenvalues obtained from a single diagonalization of the secular determinant, provides the vibrational frequency. The frequency for H_2^+ between the ground and first excited vibration states ($v = 0$ and 1) is found to be $2191.099\,52\,\text{cm}^{-1}$ which is in excellent agreement with experiment $(2191.2 \pm 0.2\,\text{cm}^{-1})$,^[129] and the theoretical calculations $(2191.099\,519\,\text{cm}^{-1})$ ^[30]. If the BO approximation is assumed, the vibrational frequencies need to be calculated using the potential energy surface. This is undesirable due to the

coupling of electronic and nuclear motions being neglected resulting in approximate values, further highlighting the advantages of treating a molecule non-adiabatically.

Table 4.1 The electronic energy, E (a.u.), dissociation energy, D_0 (cm^{-1}), and bond-length, r (a.u.) for the non-adiabatic and Born-Oppenheimer electronic hydrogen molecule isotopologues ($m_e = 1, \hbar = 1, e = 1$), and the muonic energy, dissociation energy and bond-length in muon-atomic units ($m_\mu = 1, \hbar = 1, e = 1$). A wavefunction corresponding to $\omega = 30$ (see text for details) was used to obtain the non-adiabatic data.

	Non-adiabatic			Born-Oppenheimer			Experiment	
	Energy, E ¹	$\langle r_3 \rangle^2$	D_0	$E_{BO} + \text{ZPE}$	r_{BO}	$D_0(\text{BO})$	r_e^3	D_0 (expt)
$\mu^+ \mu^+ e^-$	-0.585 126 098 25	2.205 215 237	19 211.19					
$p^+ p^+ e^-$	-0.597 139 063 07	2.063 913 867	21 379.29	-0.597 341	1.997 441	21363.87	1.987 99	21379.37(8) ²
$d^+ d^+ e^-$	-0.598 788 784	2.044 070 029	21 711.52	-0.598 889	1.997 441	21703.62	1.995 36	21711.64(7) ³
$t^+ t^+ e^-$	-0.599 506	2.035 386 699	21 859.20	-0.599 574	1.997 441	21853.96	1.996 68	-
$p^+ d^+ e^-$	-0.597 897 968 60	2.054 803 238	21 516.01	-0.598 049	1.997 441	21519.26	-	21516.12(10) ⁴
$p^+ t^+ e^-$	-0.598 176 134 6	2.051 456 621	21 567.13	-0.598 311	1.997 441	21576.76	-	-
$d^+ t^+ e^-$	-0.599 130 661	2.039 939 515	21 776.62	-0.599 214	1.997 441	21774.95	-	-
$\mu^+ \mu^+ \mu^-$	-0.262 005 070 23	8.548 580 655	544 794.80					
$p^+ p^+ \mu^-$	-0.494 386 812 860 9	3.299 486 184	2 041 793.37					
$d^+ d^+ \mu^-$	-0.531 111 130 611 07	2.834 451 765	2 621 871.16					
$t^+ t^+ \mu^-$	-0.546 374 222 033 39	2.652 824 758	2 927 037.93					
$p^+ d^+ \mu^-$	-0.512 711 790 563	3.100 710 462	1 786 901.89					
$p^+ t^+ \mu^-$	-0.519 880 084 536 7	3.036 524 320	1 724 723.57					
$d^+ t^+ \mu^-$	-0.538 594 970 881 1	2.747 914 141	2 574 012.45					

¹ All digits presented are converged. Bold digits are in agreement with^[30] for electronic systems and^[40] for muonic systems. Note, older mass data are used by^[30] and^[40].

² Bold digits in agreement with^[30] for electronic systems (except for $\mu^+ \mu^+ e^-$ which was taken from^[134]). Muonic systems were compared to data in the papers of Frolov^[134,41,135,34].

³ Taken from NIST: <http://webbook.nist.gov/chemistry/> and converted from Å to a.u. using the conversion 1 a.u. (Bohr) = 0.529 177 210 67 Å; NIST values: H_2^+ : 1.052 Å, D_2^+ : 1.0559 Å, T_2^+ : 1.0566 Å.

4.4.2 Intracule Densities

The intracule density, $h(r)$, represented in inter-particle coordinates is^[136–138]

$$h(r) = \int_0^\infty \int_0^\infty \int_{|r_1-r_2|}^{r_1+r_2} \Psi(r_1, r_2, r_3) \delta(r_3 - r) \Psi(r_2, r_2, r_3) dr_3 dr_2 dr_1. \quad (4.28)$$

The intracule densities, $h(r)$ and radial intracule densities $4\pi r^2 h(r)$ for both the electronic and muonic homonuclear and heteronuclear molecular systems are shown in Figure 4.4.

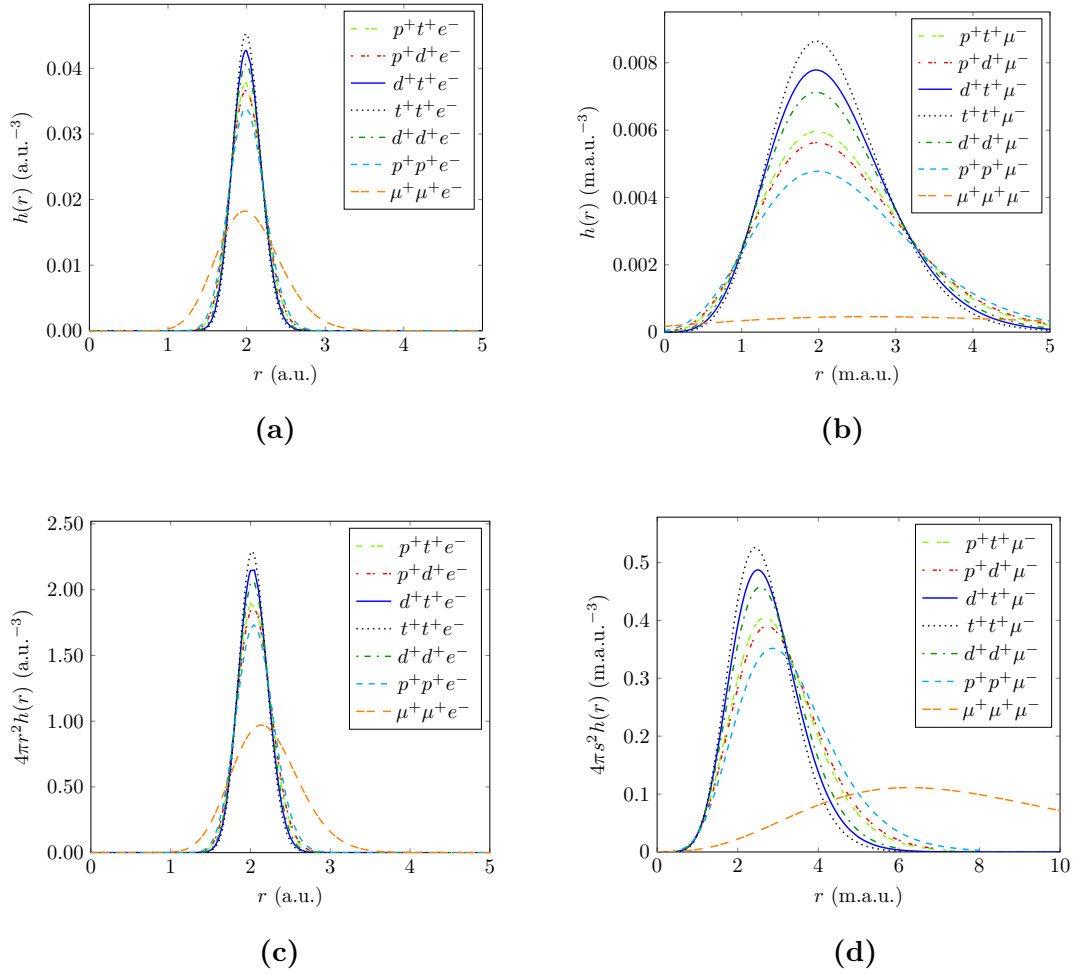


Figure 4.4 Intracule densities $h(r)$ (a) for electronic systems, (b) for muonic systems and radial intracule densities $4\pi r^2 h(r)$ (c) for electronic systems and (d) muonic systems. Electronic systems measured in atomic units and muonic systems measured in muon-atomic units, where $1 \text{ m.a.u.} = \frac{1}{206.7682826} \text{ a.u.}$

The intracule densities for the symmetric systems ($a_1^+ = a_2^+$) are in excellent agreement with those available in the literature^[19]. Arias *et al.*^[20] studied non-

symmetric systems in a later paper but did not present intracule densities, which have been plotted in this work, Figure 4.4.

The key features of the intracule densities (the maximum in the distribution r_{max} , the density at the maximum point $h(r_{max})$, and the FWHM for each spatial distribution) are provided in Table 4.2. Also provided (in italics) are the data for the radial intracule distributions.

The maximum in the density occurs at $r_{max} \approx 2$ a.u. for the electronic systems, i.e., close to the Born Oppenheimer result, and $r_{max} = 2/m_\mu = 9.672 \times 10^{-3}$ a.u. for the muonic systems, with the exception of $\{\mu^+\mu^+\mu^-\}$ which peaks at a much greater distance (Table 4.2). In line with the principles of muon catalysed fusion^[139,140], the nucleus-nucleus bond length decreases significantly when the electron is replaced by the heavier muon.

The mass ratio a_i/a_3 , $i = (1, 2)$, controls the localisation of the intracule densities, and as the ratio a_i/a_3 gets smaller, i.e., $a_i \rightarrow t^+ \rightarrow d^+ \rightarrow p^+ \rightarrow \mu^+$, the intracule densities become more diffuse for both the electronic and muonic systems. This “uncertainty” in the internuclear distance (Fig. 4.4) is a manifestation of the vibrational motion in the BO picture. The distribution is approximately symmetric about the maximum in the distribution, due to the non-BO coupling of the electronic and nuclear motion, and the inherent anharmonicity of the vibrational motion resulting in a non-symmetric distribution.

Table 4.2 Key features of the intracule densities $h(r)$ provided in Fig. 4.4 (values in italics correspond to the radial intracule distribution: $4\pi r^2 h(r)$). Values in atomic units (above thick line) and muon-atomic units (below thick line).

	r_{max}	$h(r_{max})$	FWHM
$\mu^+ \mu^+ e^-$	1.977 798	0.018 245	0.940 547
	<i>2.136 628</i>	<i>0.970 123</i>	<i>0.966 356</i>
$p^+ p^+ e^-$	1.989 685	0.033 910	0.537 419
	<i>2.041 771</i>	<i>1.731 407</i>	<i>0.542 240</i>
$d^+ d^+ e^-$	1.991 768	0.040 740	0.451 292
	<i>2.028 470</i>	2.068 576	<i>0.457 632</i>
$t^+ t^+ e^-$	1.992 429	0.045 245	0.407 779
	<i>2.022 506</i>	<i>2.291 297</i>	<i>0.410 064</i>
$p^+ d^+ e^-$	1.990 635	0.036 610	0.499 693
	<i>2.035 674</i>	<i>1.864 502</i>	<i>0.533 209</i>
$p^+ t^+ e^-$	1.989 456	0.037 749	0.485 305
	<i>2.032 119</i>	<i>1.917 962</i>	<i>0.489 938</i>
$d^+ t^+ e^-$	1.992 169	0.042 725	0.430 491
	<i>2.025 663</i>	<i>2.166 764</i>	<i>0.474 519</i>
$\mu^+ \mu^+ \mu^-$	2.632 688	0.000 459	5.870 069
	<i>6.263 763</i>	<i>0.111 155</i>	<i>8.044 725</i>
$p^+ p^+ \mu^-$	1.984 563	0.004 779	2.299 054
	<i>2.867 104</i>	<i>0.351 625</i>	<i>2.631 513</i>
$d^+ d^+ \mu^-$	1.967 179	0.007 125	1.849 166
	<i>2.559 161</i>	<i>0.457 461</i>	<i>2.034 267</i>
$t^+ t^+ \mu^-$	1.965 083	0.008 645	1.641 984
	<i>2.437 699</i>	<i>0.525 613</i>	<i>1.774 437</i>
$p^+ d^+ \mu^-$	1.977 355	0.005 630	2.101 476
	<i>2.728 007</i>	<i>0.389 979</i>	<i>2.373 243</i>
$p^+ t^+ \mu^-$	1.976 818	0.005 966	2.029 924
	<i>2.679 483</i>	<i>0.405 077</i>	<i>2.283 647</i>
$d^+ t^+ \mu^-$	1.966 082	0.007 787	1.752 309
	<i>2.500 945</i>	<i>0.487 131</i>	<i>1.912 104</i>

4.4.3 Centre of Mass Particle Densities

4.4.3.1 Homonuclear systems

Equation (4.19) was used to calculate the COM densities. Three dimensional, spherically-symmetric particle densities have been reported previously for H_2^+ ^[62] which has been done here along with other one-electron homonuclear systems. This was done by expressing the centre of mass distance s_i as an effective s_x and s_y component in order to plot the particle density in the s_x - s_y plane. Figure 4.5 shows plots of the centre of mass density, $\rho_{c,a_i}(s)$ for the electronic and muonic homonuclear diatomic molecules.

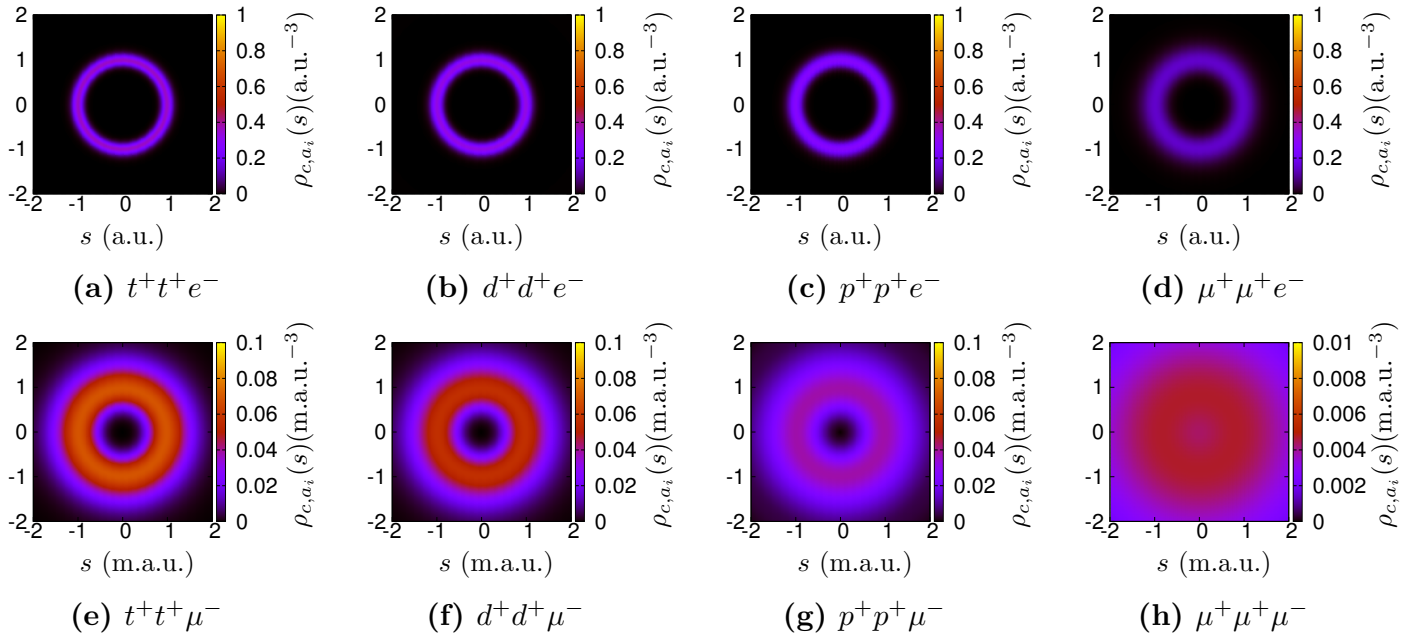


Figure 4.5 Density plots of $\rho_{c,a_i}(s)$ for $s = (s_x, s_y, 0)$ in the case of homonuclear systems where $a_1^+ = a_2^+$. The density scale is given on the right-hand side colour bar. The centre of each plot corresponds to the centre of mass. (Note scale, 1 m.a.u. = $\frac{1}{206.7682826}$ a.u.)

The top row in Figure 4.5 represents the electronic homonuclear systems and the bottom row represents the muonic homonuclear systems, and for all plots the centre of mass is at the centre of the plot. Moving from the figures on the left hand side to those on the right, the mass of the heavy particles a_1^+, a_2^+ decreases through $t^+ \rightarrow d^+ \rightarrow p^+ \rightarrow \mu^+$; and for both electronic and muonic systems it is clear that the like-charged particles are found in a shell, of finite width, at a given distance from the centre of mass. In these systems a single shell arises and the width of the shell increases as the mass of the heavy particles decreases to the point at which the zero-density at the centre of mass (centre of plot) disappears in the

$\{\mu^+\mu^+\mu^-\}$ system, Figure 4.5 (h), where all masses are equal. The colour scale represents the density distribution with the light colour highlighting the maximum in the distribution.

Figure 4.6 show slices of the three-dimensional spherically symmetric particle density relative to the centre of mass in the homonuclear ions.

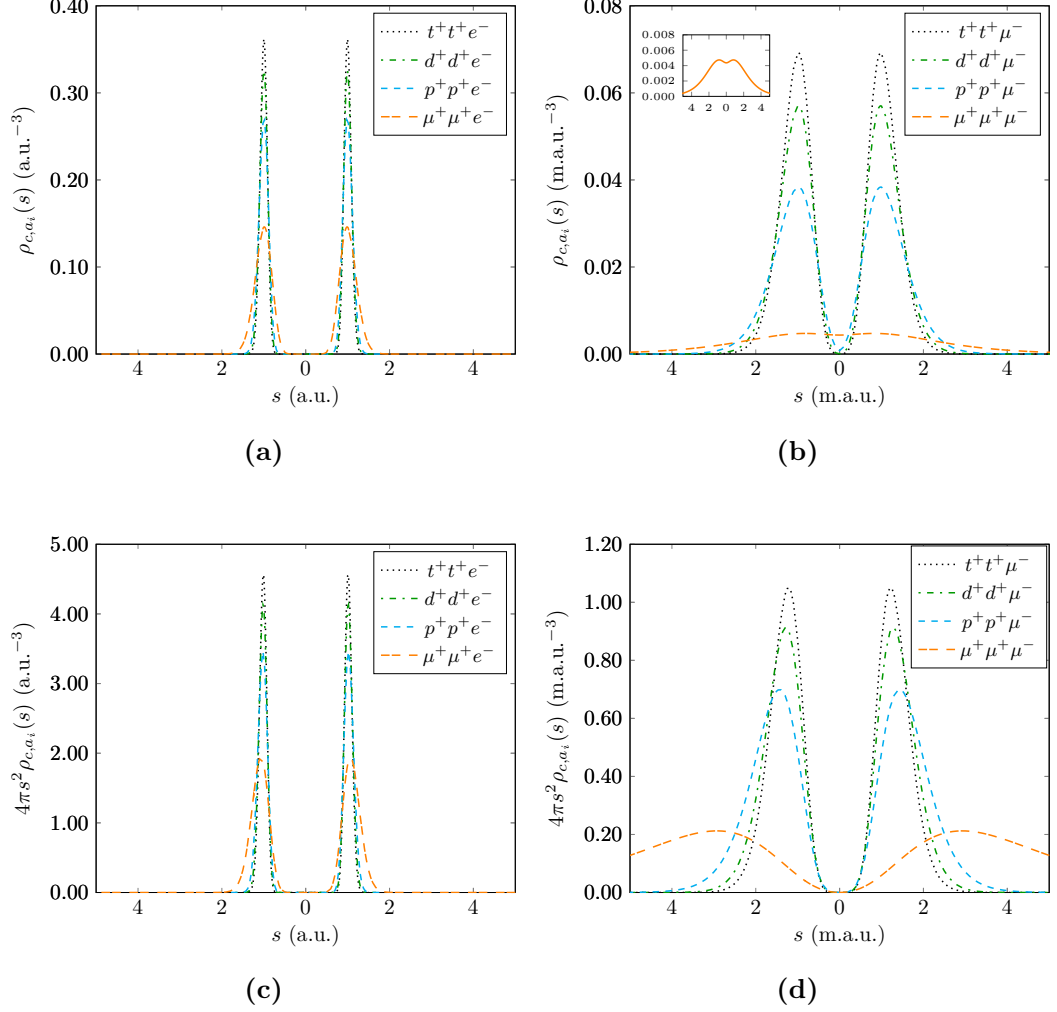


Figure 4.6 centre of mass particle densities, $\rho_{c,a_i}(s)$, for homonuclear diatomic ions, for (a) electronic systems and (b) muonic systems and radial centre of mass particle density distributions, $4\pi s^2 \rho_{c,a_i}(s)$, for (c) electronic systems and (d) muonic systems. For (c) and (d) the area under each peak is equal to 1. The inset in b) corresponds to $\{\mu^+\mu^+\mu^-\}$. The centre of mass coincides with the origin and the left peak corresponds to $\rho_{c,a_1}(s)$ and the right peak $\rho_{c,a_2}(s)$ where $m_{a_1} = m_{a_2}$. (Note scale, $1 \text{ m.a.u.} = \frac{1}{206.7682826} \text{ a.u.}$).

In Figure 4.6 the centre of mass is located at the origin of each plot. The peak on the left of the centre of mass corresponds to heavy particle a_1^+ and the peak on the right corresponds to heavy particle a_2^+ (where $m_{a_1} = m_{a_2}$). Figures 4.6 (a) and

(b) show the centre of mass particle densities $\rho_{c,a_i}(s)$ and (c) and (d) show the radial centre of mass particle densities, $4\pi s^2 \rho_{c,a_i}(s)$. The radial densities represent the case where the spherical average ($4\pi s^2$) has been applied. As the area under each of the peaks is normalised, direct comparison can be made between the different systems.

It can be seen that the localisation of the particle density is governed by the mass of the particles. As $a_1^+ = a_2^+ \rightarrow t^+ \rightarrow d^+ \rightarrow p^+ \rightarrow \mu^+$ the width of the centre of mass particle density peak increases showing a greater spatial delocalisation of the particles. This shows that as the mass becomes heavier the uncertainty in the particle's position decreases due to the reducing effects of the lighter particle on its position. The finite width of the centre of mass particle density has been attributed to the zero-point vibration of the nuclei in the Born-Oppenheimer treatment^[62]. The centre of mass particle densities show that the distribution of the like-charged particles is effectively zero at the centre of mass which is intuitive: an electron has minimal effect on the heavy particle's position. As the mass of the uniquely charged particle a_3 increases (i.e. an electron is replaced with a muon), a minimum at the centre of mass appears which can be seen in Figure 4.6 (b). When all three of the masses are equal $a_1^+ = a_2^+ = a_3^-$ it can be seen in the inset of Figure 4.6 (b) that a nonzero minimum in the heavy particle density is very apparent. This may indicate a transition from a molecule-like to an atom-like system due to mass scale similarity and the charge-inversion invariance of the Coulomb Hamiltonian, which has been shown previously for the positronium negative ion, Ps^- ^[62].

4.4.3.2 Heteronuclear systems

Shell-like densities were also calculated for the heteronuclear systems and can be seen in Figure 4.7.

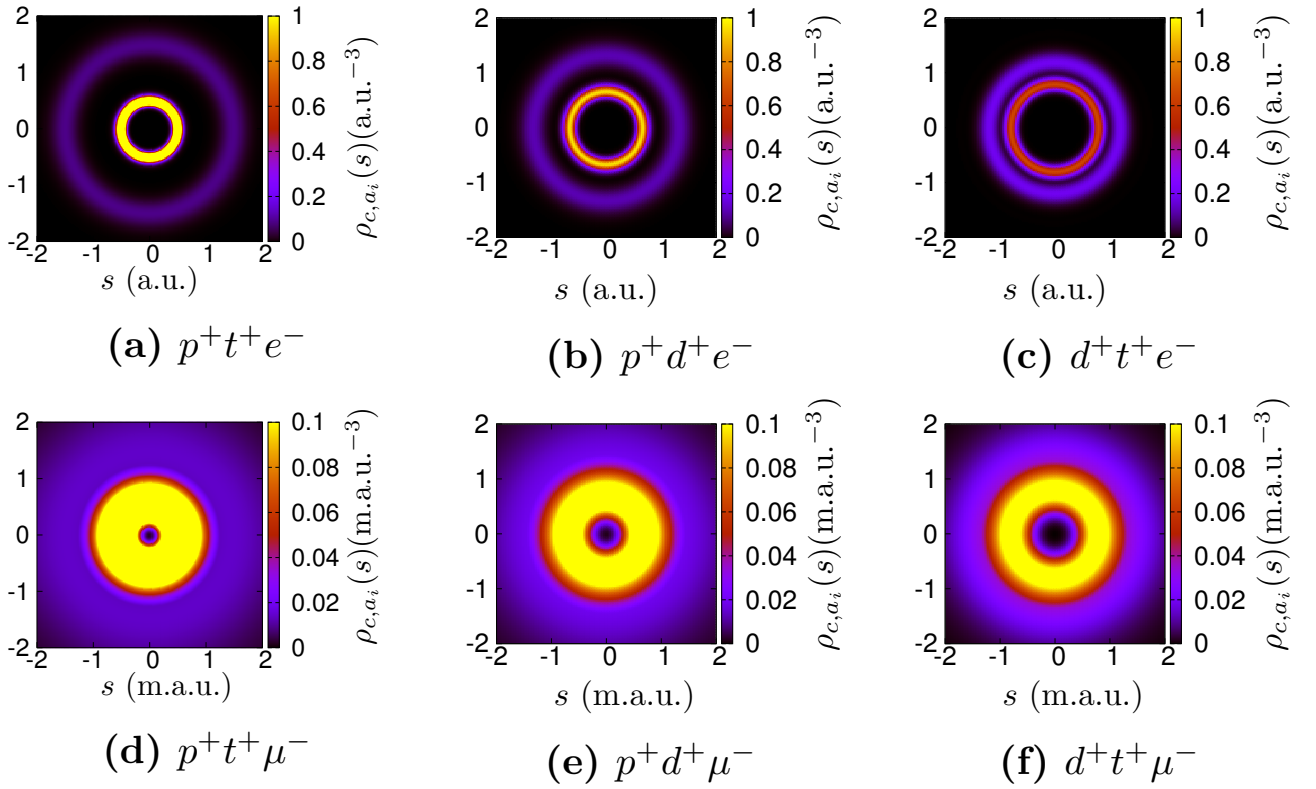


Figure 4.7 Density plots of $\rho_{c,a_i}(s)$ for $s = (s_x, s_y, 0)$ in the case of systems where $a_1^+ \neq a_2^+$. The density scale is given on the right-hand side colour bar. The centre of each plot corresponds to the centre of mass. (Note scale, $1 \text{ m.a.u.} = \frac{1}{206.7682826} \text{ a.u.}$).

The heteronuclear centre of mass particle densities show a distinctly different behaviour to that in the homonuclear case. As before the centre of the plot is the position of the centre of mass. The distance of the particle shell from the centre of mass and the width of the shell are dependent on the relative masses of the particles. For the one electron systems there are two distinct particle shells. In Figure 4.7 the mass ratio m_2^+/m_1^+ decreases going from the figures on the left to those on the right. The inner particle shell is closer to the centre of mass, which represents the heavier particle in each particle pair. As the inverse mass ratio decreases it can be seen that the radius of the inner shell increases whereas the outer particle shell decreases in radius, signifying a change in the centre of mass position caused by the decreasing mass ratio $m_{\text{light}}/m_{\text{heavy}}$. As the mass of the lighter particle increases the centre of mass position is located closer to the increasingly heavier particle. For the muonic systems the shells merge showing an overlap in their particle densities. There is a higher probability of them occupying the same region of space.

In Figure 4.8, slices of the three-dimensional spherically-symmetric centre of mass particle densities are presented for the heteronuclear diatomic ions.

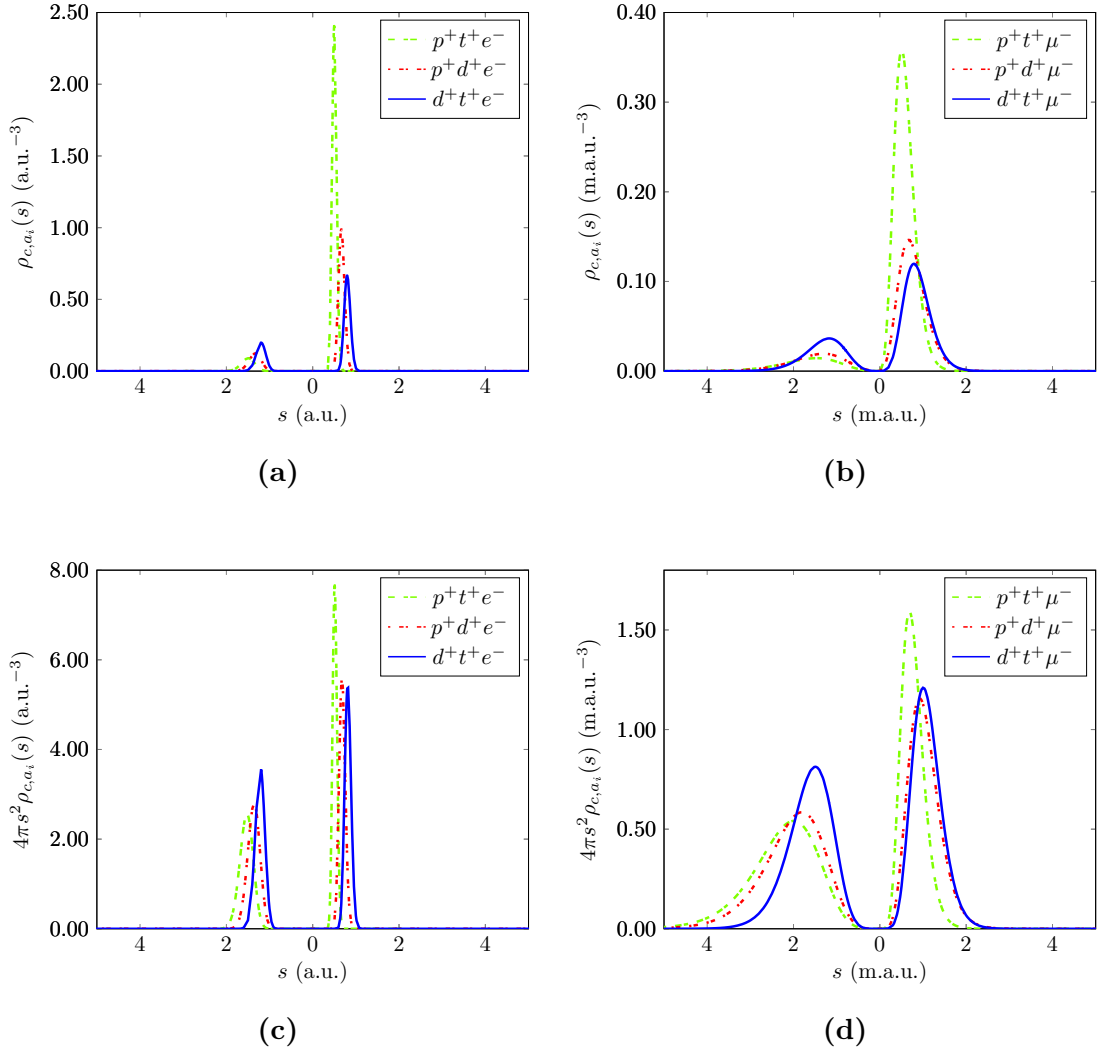


Figure 4.8 For heteronuclear diatomic ions (a) centre of mass particle densities, $\rho_{c,a_i}(s)$, for electronic systems and (b) for muonic systems, (c) radial center of mass particle density distributions $4\pi s^2 \rho_{c,a_i}(s)$ for electronic systems and (d) radial center of mass particle density distributions for muonic systems; for (c) and (d) the area under each peak is equal to 1. The centre of mass coincides with the origin and the left peak corresponds to $\rho_{c,a_1}(s)$ and the right peak $\rho_{c,a_2}(s)$ where in each case

$$m_{a_1} \leq m_{a_2}. \quad (\text{Note scale, } 1 \text{ m.a.u.} = \frac{1}{206.7682826} \text{ a.u.}).$$

The key features of the plots in Figure 4.8 have been provided in Table 4.3. Due to the negligible effect of the electron mass on the much heavier nuclei, the peak-to-peak distance $(s_{1_{max}} + s_{2_{max}})$ of the centre of mass particle density plots is in excellent agreement with the r_{max} positions from the radial intracule density plots given in Figure 4.4 and Table 4.3. This shows that the centre of mass position lies almost on the inter-nuclear distance r_3 , as the electron cannot move the centre of mass position towards itself due to its significantly smaller mass.

Whilst this is true for the electronic systems, the muonic peak-to-peak distance from the centre of mass plots is not in such good agreement with the muonic intracule densities. This is due to the heavier muon which means the mass ratio m_i/m_3 is smaller and the muon moves the centre of mass position away from r_3 and thus the Born-Oppenheimer separation is less appropriate and the spatial distributions are more diffuse.

The mass-weighted coordinate displacements along the internuclear axis in the BO picture show a mass-ratio dependence. As an example consider again the $d^+t^+e^-$ system which has vibrational displacements (calculated from a frequency calculation) of 0.83 for the deuteron and -0.56 for the triton. This shows that the ratio of the displacements is inversely proportional to their mass ratio, $0.83/0.56 = 1.4821$ and $m_t/m_d = 1.4976$. The ratio $s_{1\max}/s_{2\max}$ calculated in this work is 1.4975.

As for the homonuclear systems the Full Width at Half Maximum (FWHM) of the centre of mass peaks show that the localisation of the particle density is governed by the mass of the particles. As $a_1^+ = a_2^+ \rightarrow t^+ \rightarrow d^+ \rightarrow p^+ \rightarrow \mu^+$ the FWHM increases.

In $d^+t^+e^-$ the deuteron is the lighter of the nuclei whereas in $p^+d^+e^-$ it is the heavier. Figure 4.9 shows the spatial localisation of the deuteron in each system is dependent on its partner mass (t^+ or p^+) rather than its absolute mass. The heavier mass particle is more localised, and the lighter the partner mass, the greater this localisation will be.

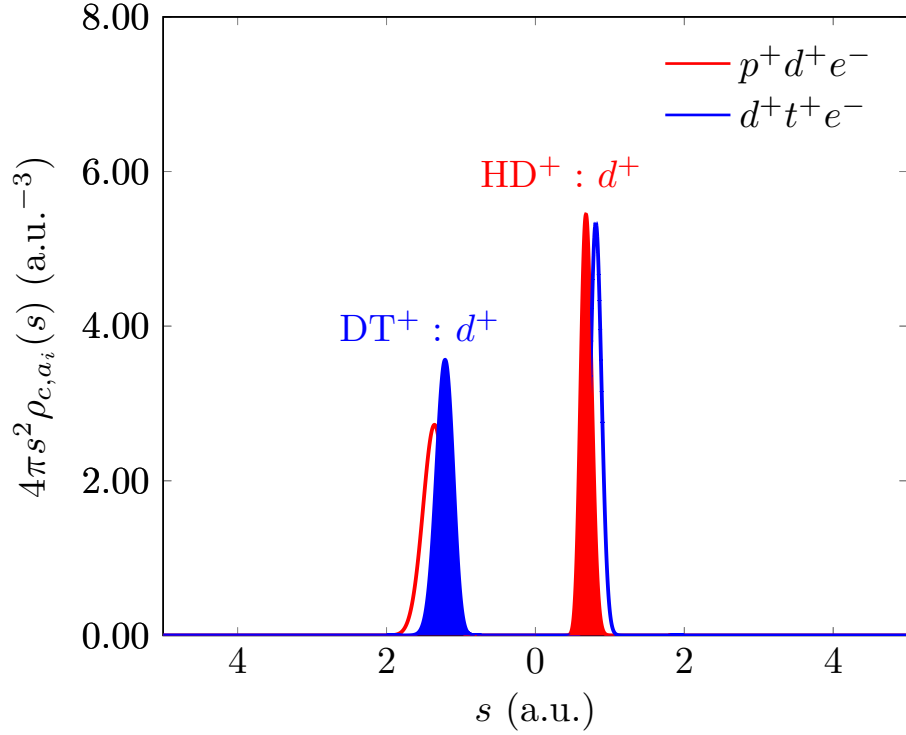


Figure 4.9 Localisation of the deuteron radial centre of mass particle density distribution in $p^+d^+e^-$ and $d^+t^+e^-$. The shaded peak represents the deuteron in each system.

Furthermore, for the fully correlated heteronuclear centre of mass particle density distributions, the ratio of $s_{1_{max}}$ to $s_{2_{max}}$ is directly proportional to their mass ratio. Using the example of $d^+t^+e^-$, $\rho_{c,d^+}(s_{max}) = 3.3489$ and $\rho_{c,t^+}(s_{max}) = 5.0152$, and the ratio of these peak heights is 0.6678 which is in excellent agreement with $m_d/m_t = 0.6677$ seen in Figure 4.10(a). Additionally the ratio of the FWHMs of the heteronuclear centre of mass particle density peaks is inversely proportional to the mass ratio m_{a_1}/m_{a_2} . These values can be seen in Table 4.3. For example, consider the $d^+t^+e^-$ system, where the FWHM of the $\rho_{c,d^+}(s)$ peak is 0.2846 and $\rho_{c,t^+}(s)$ is 0.1900. This gives a ratio of FWHM's of 1.4975 which compared to the inverse mass ratio $m_t/m_d = 1.4976$ is in very good agreement and can be seen in Figure 4.10(b).

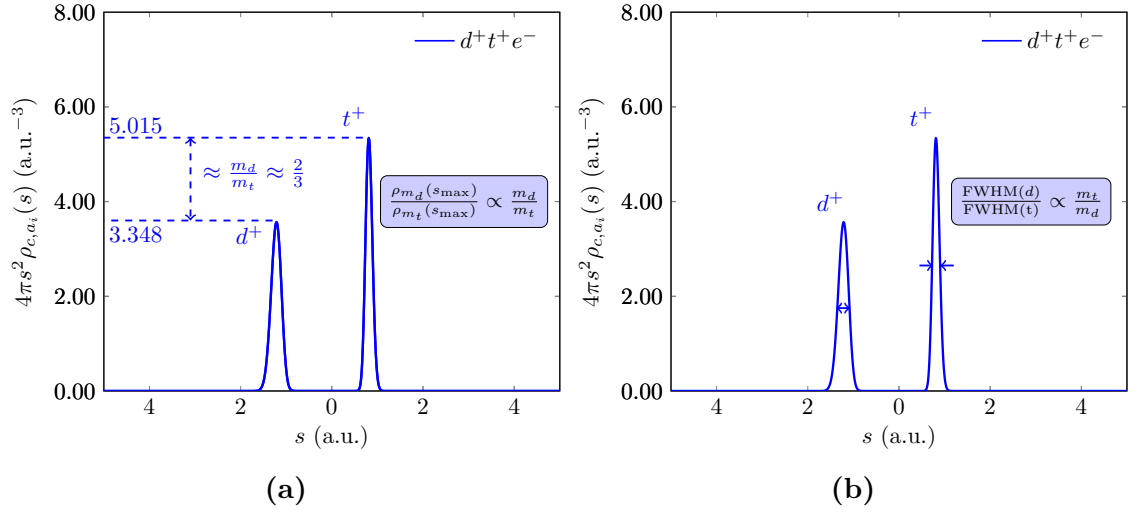


Figure 4.10 Radial centre of mass particle densities for $d^+t^+e^-$ highlighting (a) the ratio of the probability maximum for d^+ and t^+ in $d^+t^+e^-$, and (b) the ratio of the full-width at half maximum (FWHM) for d^+ and t^+ in $d^+t^+e^-$.

When assuming the BO approximation these structural features and mass ratio dependencies are lost due to the nuclei being fixed.

The relations found for FWHM and peak heights hold well for the electronic systems but also hold true for muonic systems, albeit with a lesser agreement which can be seen in Table 4.3. This is caused by a greater coupling between the nuclear and muonic motion than in the electronic case.

Table 4.3 The key features of the radial center of mass particle density distributions provided in Figs. 4.6 and 4.8 where s_1 corresponds to the lighter particle and s_2 the heavier particle in the heteronuclear diatomic ions $\{a_1^+ a_2^+ a_3^-\}$. Values in atomic units (above thick line) and muon-atomic units (below thick line).

	$s_{1,max}$	$\rho(s_{1,max})$	FWHM (s_1)	$s_{2,max}$	$\rho(s_{2,max})$	FWHM (s_2)	$s_{1,max} + s_{2,max}$
$\mu^+ \mu^+ e^-$	1.068 309	1.940 095	0.483 222	1.068 309	1.940 095	0.483 222	2.136 618
$p^+ p^+ e^-$	1.020 850	3.462 470	0.271 140	1.020 650	3.462 470	1.020 650	2.041 706
$d^+ d^+ e^-$	1.013 338	4.109 394	0.228 804	1.013 338	4.109 394	0.228 804	2.026 676
$t^+ t^+ e^-$	1.007 564	4.570 708	0.205 031	1.007 564	4.570 708	0.205 031	2.022 506
$p^+ d^+ e^-$	1.356 830	2.674 921	0.355 374	0.678 844	5.346 461	0.177 804	2.035 674
$p^+ t^+ e^-$	1.523 221	2.558 737	0.367 245	0.508 898	7.658 750	0.122 704	2.032 119
$d^+ t^+ e^-$	1.214 572	3.348 956	0.284 560	0.811 063	5.015 176	0.190 021	2.025 636
$\mu^+ \mu^+ \mu^-$	2.25	0.191 884	4.363 512	2.25	0.191 884	4.363 512	4.5
$p^+ p^+ \mu^-$	1.430 745	0.698 989	1.322 379	1.430 745	0.698 989	1.322 379	2.861 490
$d^+ d^+ \mu^-$	1.278 907	0.912 626	1.019 387	1.278 907	0.912 626	1.019 387	2.557 815
$t^+ t^+ \mu^-$	1.218 556	1.049 694	0.886 028	1.218 556	1.049 694	0.886 028	2.437 113
$p^+ d^+ \mu^-$	1.803 231	0.585 048	1.579 223	0.922 087	1.156 499	0.801 813	2.725 318
$p^+ t^+ \mu^-$	1.993 417	0.541 311	1.706 660	0.686 117	1.592 155	0.582 556	2.679 534
$d^+ t^+ \mu^-$	1.494 234	0.812 923	1.145 328	1.005 823	1.209 780	0.770 012	2.500 058

4.5 Summary and Conclusions

Using a high accuracy fully correlated wavefunction, and not assuming the Born-Oppenheimer approximation, nuclear motion has been studied in three particle homonuclear and heteronuclear diatomics with the use of intracule and centre of mass particle densities. A new coordinate system was developed in order to preserve the inter-particle coordinate system of the wavefunction whilst allowing for an explicit definition of the centre of mass position from any of the three particles. The results show that as the mass of the heavy particles a_1^+, a_2^+ increase, the intracule and centre of mass particle density distributions become more localised showing a decrease in the uncertainty in their position. For the first time, centre of mass particle densities were used to study the motion of the heavy masses in heteronuclear diatomics revealing interesting insight into the correlated effects of nuclear motion in quantum mechanics.

It has been found that the spatial localisation of each nucleus can be quantified in terms of the FWHM and $\rho_{c,a_i}(s_{max})$. The ratio of the FWHM was found to be directly proportional to the inverse mass ratio m_{a_2}/m_{a_1} and the ratio of the peak heights $\rho_{c,a_1}(s_{max})/\rho_{c,a_2}(s_{max})$ is directly proportional to the mass ratio m_{a_1}/m_{a_2} .

To offer greater insight into the systems studied in this work, BO results were calculated for the electronic systems using Gaussian09 using HF and a (very large) aug-cc-pV6Z basis set. Equilibrium distances, r_3 , were calculated along with dissociation energies for both the fully correlated and BO methodologies and it was found that despite being an approximation, the BO approach still provides accurate results for these molecular diatomics. A similar mass ratio dependence behaviour was found within the BO picture with mass-weighted displacements along the internuclear axis. However, the max in $\rho_{c,a_i}(s_{max})$ are not available within a BO treatment.

Future work could involve trying to relate the zero-point vibration to the widths of the intracule and centre of mass particle densities, based upon the hypothesis that the finite width of the centre of mass particle density is related to the zero-point vibration^[62]. Preliminary investigation by the author of this thesis has found that no simple, direct relation is achievable to match the width to the zero-point vibration. This work was only concerned with the ground state of S-state systems, but it would be interesting to use the same methodology on excited S and P states to see how it affects the centre of mass particle densities and, to determine if the new correlations reported in this chapter still hold.

CHAPTER 5



Electron correlation: energies, densities and Coulomb holes

5.1 Introduction

Electron correlation is the instantaneous interaction between two electrons in a quantum system. The equations used in quantum mechanics cannot be solved exactly for many-particle systems and thus electron correlation remains one of the biggest challenges in quantum physics and chemistry. In quantum chemistry, quantum mechanics is applied to understand how electrons interact with nuclei and with each other.

Three-particle systems offer the smallest quantum systems in which electron correlation can be studied. The nucleus of the atom attracts the two electrons, whilst simultaneously the electrons try to repel one another, which is the complex effect we are looking to study. There is a vast history of research into this subject area and constant advancement in the present day to develop new methods to reduce this complex effect to a more manageable form. Quantum mechanics is almost 100 years old and still, only simple one electron atomic systems can be treated exactly. The helium atom was the first three-particle system to yield an approximate wave function seen in the pioneering work of Hylleraas in 1929 whose calculation of the ionisation energy of helium was accurate to within 0.01eV of experiment^[7]. The techniques developed by Hylleraas do not scale well for systems with more than two electrons hence more suitable methods have been developed with a key example being Hartree-Fock (HF) theory which forms the bedrock of conventional quantum chemistry.

There are two main effects that electron correlation has on a quantum system; the first being the Coulomb hole where electrons repel one another at small distances due to their like-charge causing a small decrease in the probability of the electrons being found near at small separation. The second effect is the Fermi hole where electrons of the same spin cannot be found in the same spatial region due to the Pauli exclusion principle and is thus a purely quantum mechanical phenomenon. The Coulomb hole was first studied by Coulson and Nielsen in 1961^[141] using a Hylleraas wavefunction,

to calculate intracule densities and directly comparing to the HF intracule densities where the electron correlation is “switched off”. They concluded that the Coulomb hole is $1.1a_0$ wide and moves $0.047e$ of charge (area of the hole). Since then, there has been more investigation into the Coulomb hole and evidence of a secondary Coulomb hole for helium has been presented^[142,137] however it was dismissed as a numerical artefact from inaccurate calculations of the particle density caused by a poor basis. More recently Pearson *et al.* showed that the secondary Coulomb hole for helium emerges as the quality of the basis improves, highlighting it is a feature of a good basis, not a bad one^[143].

Fermi correlation was studied by Maslen in 1956 who calculated the size of the Fermi hole to be $-1e$ along with its topological features and found that the Fermi hole has a significant effect on the energy of a system^[144,145], building on the idea of Slater who had just introduced the concept of an averaged exchange hole as a basis for the approximation of the exchange potential^[146], due to the anti-symmetry principle as electrons are fermions.

Löwdin defined Coulomb electron correlation as the difference between the energy calculated using a method where the electron correlation is explicitly included, which here is described as a fully correlated (FC) method, and one without electron correlation^[147,148], i.e. Hartree-Fock theory (HF) which still correctly accounts for the effects of spin. The difference in these energies can be written as

$$E_{\text{corr}} = E - E_{\text{HF}} \quad (5.1)$$

Where E is the exact, non-relativistic, fully correlated energy and E_{HF} is the HF energy. The correlation energy is always negative as the HF energy is an upper bound to E .

Hartree-Fock theory treats electron-electron interactions using a mean-field approach which means each electron moves in a field generated by the other electron(s), thus their motions are statistically independent. Many problems in quantum mechanics involve atoms and molecules which have a large number of electrons around a number of nuclei hence HF can offer insight into these systems without the computational cost of explicitly including the electron correlation. HF theory is also used as a reference wavefunction for electron-correlated methods referred to as post-HF methods, with examples including coupled cluster theory and configuration interaction^[68]. It has also contributed to developments in approximate density functional theory (DFT), where hybrid exchange-correlation (XC) functionals include some “exact” HF exchange (X). Additionally, there are various correlation (C) functionals used in DFT which have their basis in HF such as the LYP correlation functional^[149]

along with subsequent functional developments e.g. ^[150,151]. These are based on the second order reduced density matrix and exponential correlation factor following from the approach used by Colle and Salvetti ^[152].

The effect of electron correlation is most dominant for a two-electron system just prior to electron detachment. The critical nuclear charge for a two-electron atom, Z_C , has previously been determined for a fully correlated method using high accuracy variational calculations ^[94], with the most accurate value calculated being $Z_C = 0.911\ 028\ 224\ 077\ 255$ ^[47].

In order to develop a new correlation functional, very high accuracy electron correlation data is required, especially the long-range behaviour of electron motion. Previous chapters have calculated Fully Correlated (FC) energies, whereas this chapter is dedicated to calculating Hartree Fock (HF) energies which can then be compared against the FC energies to study electron correlation.

Chapter summary

This chapter will describe in detail the implementation of a new Hartree-Fock method which uses a Laguerre-based wavefunction. This basis was chosen in order to investigate electron correlation in two-electron atomic systems by comparison with the FC method described in Chapter 2.

The FC and HF methods are used to study the effects of electron correlation on physical properties using particle densities which highlights the presence of Coulomb hole(s) for the helium isoelectronic sequence along with high accuracy data in the low density regime. Additionally, the critical nuclear charge for binding two electrons within the HF methodology is calculated and the behaviour of the system near this critical value is investigated. The Hartree-Fock implementation wavefunction and energy results discussed in this chapter have been published¹ in Philosophical Transactions of the Royal Society A ^[153] and the Coulomb hole results are in the process of being submitted for publication.

5.2 Method

5.2.1 Schrödinger equation and choice of coordinates

In this work the nucleus is fixed, meaning the two-electron Schrödinger equation in atomic units has the form

¹Royal Society gives permission for published content to be re-used in derivative works, www.royalsociety.org/journals/permissions/

$$\left(-\frac{1}{2}\nabla_1^2 - \frac{1}{2}\nabla_2^2 - \frac{Z}{r_1} - \frac{Z}{r_2} + \frac{1}{r_{12}}\right)\Psi = E\Psi, \quad (5.2)$$

For atoms, it is usual for a solution to be constructed for the translation-free problem directly and the angular and radial coordinates are integrated over. The work conducted here focuses on ground states, therefore the angular terms integrate to give a constant factor and the angular momentum is separated off so a body-fixed problem is solved as in the FC method, Chapter 2. The three internal coordinates are chosen to be the interparticle distances, r_1, r_2, r_{12} , hence the problem requires integration over r_{12} when conventionally HF does not as the electron-electron distance is treated as the difference between the nuclear-electron components r_1, r_2 ^[154]. In this work $dr_1 dr_2 dr_{12}$ is integrated over as it facilitates an easy conversion to perimetric coordinates and the r_{12} term in the Jacobian conveniently cancels with the $1/r_{12}$ term in the two-electron integrals. Two methods are now compared, 1) following the conventional means of integrating over the angular and radial coordinates, $\theta_1, \phi_1, \theta_2, \phi_2, r_1, r_2$ ^[154] and 2) integrating over the radial coordinates only, r_1, r_2, r_{12} adhered to in this work.

1) Integrating over $\theta_1, \phi_1, \theta_2, \phi_2, r_1, r_2$

The Coulomb integral for the configuration $1s^2$ of an atom of atomic number Z can be represented for the case where $r_1 > r_2$ using the following expansion^[154]

$$\frac{1}{r_{12}} = \frac{1}{r_1} \sum_{l, m_l} \left(\frac{4\pi}{2l+1}\right) \left(\frac{r_2}{r_1}\right)^l Y_{l, m_l}^*(\theta_1, \phi_1) Y_{l, m_l}(\theta_2, \phi_2), \quad (5.3)$$

and when $r_2 > r_1$ the coordinates r_1 and r_2 are interchanged. The trial wavefunction used in this example to evaluate the Coulomb integral has the form (in atomic units)

$$\Psi = \left(\frac{Z^3}{\pi}\right)^{\frac{1}{2}} e^{-Z(r_1+r_2)}. \quad (5.4)$$

The constant factor, $(Z^3/\pi)^{1/2}$, is not carried through the derivations to follow in order to make them clearer. For the configuration under consideration the wavefunction is independent of the angular components which simplifies the problem. The integration over all six coordinates takes the form

$$\left\langle \frac{1}{r_{12}} \right\rangle = \int_0^{2\pi} \int_0^{2\pi} \int_0^\pi \int_0^\pi \int_0^\infty \int_0^\infty \Psi \frac{1}{r_{12}} \Psi r_1^2 r_2^2 dr_1 dr_2 \sin \theta_2 d\theta_2 \sin \theta_1 d\theta_1 d\phi_2 d\phi_1 \quad (5.5)$$

Note, due to the calculation being done in spherical polar coordinates, the inclusion of $r_1^2 \sin(\theta_1)$ and $r_2^2 \sin(\theta_2)$ are a consequence of the Jacobian when transforming from Cartesian coordinates to spherical polar coordinates. Substituting the wavefunction from equation (5.4) into equation (5.5) gives

$$\left\langle \frac{1}{r_{12}} \right\rangle = \underbrace{\int_0^{2\pi} d\phi_1}_{(1)} \underbrace{\int_0^{2\pi} d\phi_2}_{(2)} \underbrace{\int_0^\pi \sin \theta_1 d\theta_1}_{(3)} \underbrace{\int_0^\pi \sin \theta_2 d\theta_2}_{(4)} \int_0^\infty \int_0^\infty \frac{e^{-2Z(r_1+r_2)}}{r_{12}} r_1^2 r_2^2 dr_1 dr_2. \quad (5.6)$$

Evaluating the angular integrals (1),(2),(3),(4), gives $2\pi, 2\pi, 2, 2$ respectively, resulting in the constant angular term $16\pi^2 = (4\pi)^2$. The radial integrals are divided into two parts, one where $r_1 > r_2$ and one where $r_2 > r_1$. Integrating over the spherical harmonics in equation (5.3) gives (example given for θ_2)

$$\int_0^{2\pi} \int_0^\pi Y_{l,m_l}(\theta_2, \phi_2) \sin(\theta_2) d\theta_2 d\phi = \delta_{l,0} \delta_{m_l,0}. \quad (5.7)$$

The integration in equation (5.7) gives zero except when $l = 0$ and $m_l = 0$. Hence, the sum in equation (5.3) reduces to a single term inside the integral. When $r_1 > r_2$, $1/r_{12} = 1/r_1$ and when $r_2 > r_1$, $1/r_{12} = 1/r_2$. The integration over r_1 is split into two integrals, one for each condition, $r_2 > r_1$ and $r_1 > r_2$

$$\int_0^\infty \left\{ \underbrace{\int_0^{r_2} \frac{r_1^2 e^{-2Zr_1}}{r_2} dr_1}_{r_2 > r_1} + \underbrace{\int_{r_2}^\infty \frac{r_1^2 e^{-2Zr_1}}{r_1} dr_1}_{r_1 > r_2} \right\} r_2^2 e^{-2Zr_2} dr_2. \quad (5.8)$$

These integrals can be evaluated using the known analytical forms^[155]

$$\int_0^b x^2 e^{-ax} dx = \left(\frac{1}{a^3} \right) [2 - (2 + a^2 b^2 + 2ab)e^{-ab}], \quad (5.9)$$

and

$$\int_b^\infty x e^{-ax} dx = \left(\frac{1}{a^2} \right) [1 + ab]e^{-ab}. \quad (5.10)$$

Applying these analytical forms, combining terms and simplifying leads to the integral for the case $r_2 > r_1$

$$\int_0^{r_2} \frac{r_1^2 e^{-2Zr_1}}{r_2} dr_1 = \frac{1 + (-2Z^2 r_2^2 - 2Zr_2 - 1)e^{-2Zr_2}}{4Z^3 r_2}, \quad (5.11)$$

and for the case when $r_1 > r_2$

$$\int_{r_2}^{\infty} r_1 e^{-2Zr_1} dr_1 = \left(\frac{1}{4Z^2} + \frac{r_2}{2Z} \right) e^{-2Zr_2}. \quad (5.12)$$

Summing these results together then allows for integration over the r_2 coordinate

$$\int_0^{\infty} \left\{ \frac{1 + (-2Z^2 r_2^2 - 2Zr_2 - 1)e^{-2Zr_2}}{4Z^3 r_2} + \left(\frac{1}{4Z^2} + \frac{r_2}{2Z} \right) e^{-2Zr_2} \right\} r_2^2 e^{-2Zr_2} dr_2. \quad (5.13)$$

Upon simplification

$$- \frac{1}{4} \int_0^{\infty} \frac{Zr_2^2 e^{-4Zr_2} + r_2 e^{-4Zr_2} - r_2 e^{-2Zr_2}}{Z^3} dr_2. \quad (5.14)$$

Integration by parts gives

$$\lim_{r_2 \rightarrow \infty} \left(-\frac{1}{128} \frac{8(e^{-2Zr_2})^2 Z^2 r_2^2 + 12(e^{-2Zr_2})^2 Zr_2 - 16e^{-2Zr_2} Zr_2 + 3(e^{-2Zr_2})^2 - 8e^{-2Zr_2} - 5}{Z^5} \right) \quad (5.15)$$

Taking the limit, $r_2 \rightarrow \infty$, causes the exponential terms and their coefficients to collapse to zero leaving

$$- \frac{1}{128} \frac{-5}{Z^5} = \frac{5}{128Z^5}, \quad (5.16)$$

which is confirmed in Atkins^[154]. Multiplying by the constant angular term from equation (5.6), $(4\pi)^2$, gives

$$\left\langle \frac{1}{r_{12}} \right\rangle = \frac{5(4\pi)^2}{128Z^5} = \frac{5\pi^2}{8Z^5}. \quad (5.17)$$

2) Integrating over r_1, r_2, r_{12}

The explicit integration over all three radial variables, r_1, r_2 and r_{12} is done in order to compare with the result from the standard integration route in method 1). The same wavefunction is used, equation (5.4), once again without the constant term at the front for simplicity

$$\left\langle \frac{1}{r_{12}} \right\rangle = \int_0^{\infty} \int_0^{\infty} \int_{|r_1 - r_2|}^{r_1 + r_2} \Psi \frac{1}{r_{12}} \Psi 8\pi^2 r_1 r_2 r_{12} dr_{12} dr_2 dr_1, \quad (5.18)$$

where the integration domain for r_{12} ranges from $|r_1 - r_2| \rightarrow r_1 + r_2$ as a consequence of the triangular condition. For methods 1) and 2) to be equivalent, integration over *just* the r_{12} component should be equivalent to integration over all the angular variables $\theta_1, \phi_1, \theta_2, \phi_2$. Substituting the wavefunction from equation (5.4) into equation (5.18) and integrating over r_{12} gives

$$\left\langle \frac{1}{r_{12}} \right\rangle = \int_0^\infty \int_0^\infty 8 \left(e^{-Z(r_1+r_2)} \right)^2 \pi^2 r_1 r_2 (r_1 + r_2 - |r_1 - r_2|) dr_2 dr_1. \quad (5.19)$$

Considering either case, $r_1 > r_2$ or $r_2 > r_1$ gives the same constant factor, i.e. consider when $r_1 > r_2$

$$\left\langle \frac{1}{r_{12}} \right\rangle = \int_0^\infty \int_0^\infty 16 \left(e^{-Z(r_1+r_2)} \right)^2 \pi^2 r_1 r_2^2 dr_2 dr_1. \quad (5.20)$$

The coefficient is $16\pi^2 = (4\pi)^2$, which is equal to the constant term formed from integrating over all the angular terms in method 1). Finally, calculating the remaining two integrals by parts gives

$$\left\langle \frac{1}{r_{12}} \right\rangle = \frac{5\pi^2}{8Z^5}. \quad (5.21)$$

Which is the same result from method 1), equation (5.17), showing that integrating over just the radial variables, r_1, r_2, r_{12} is valid and gives the same result for the Coulomb integrals as integrating over the angles using conventional mean. The inter-particle coordinates are bound by the triangular condition, however perimetric coordinates are used to circumvent this issue, even though the HF wavefunction has no explicit dependence on r_{12} .

5.2.2 The Hartree-Fock Method

For a general overview of Hartree Fock theory consult Chapter 2. In Hartree-Fock theory, every electron is treated as existing in its own space with the nuclei, and interacts with the other electrons in an average manner. There are four key terms in the HF method, with the first three being the being the standard kinetic and potential energy operators \hat{T} , \hat{V} and the overlap S . All these operators are two-body operators and do not include the interaction effects of the electrons. The fourth operator, \hat{G} , which is a part of \hat{V} , treats the electron interaction in an average way by including the Coulomb (\hat{J}) and exchange (\hat{K}) interactions

$$\hat{T}(i) = -\frac{1}{2} \left(\frac{\partial^2}{\partial r_i^2} + \frac{2}{r_i} \frac{\partial}{\partial r_i} \right) \quad (5.22)$$

$$\hat{V}(i) = -\frac{Z}{r_i} \quad (5.23)$$

$$S(i) = 1 \quad (5.24)$$

$$\hat{G}(i) = \sum_{j=1}^N 2\hat{J}_j(i) - \hat{K}_j(i), \quad i \neq j \quad (5.25)$$

for the i -th electron. The wavefunction form is taken as the following product

$$\Psi_{\text{HF}}(r_1, r_2) = \psi(r_1)\psi(r_2). \quad (5.26)$$

The component, $\psi(r_i)$ has the form

$$\psi(r_i) = e^{-\frac{1}{2}Ar_i} \sum_{q=0}^{\infty} C(q)L_q(Ar_i), \quad i = 1 \text{ or } 2, \quad (5.27)$$

where Laguerre functions, $e^{-x/2}L_q(x)$ are being used as the basis in a similar fashion to the FC work which allows for a more balanced description of electron correlation effects. There is a single non-linear variational parameter (NLP), A , which is optimised in order to improve the rate of convergence.

For a closed-shell system of $2n$ -electrons the effective one-electron Fock operator, \hat{F} , has the form

$$\hat{F} = \hat{H}^{\text{core}}(1) + \sum_{j=1}^n [2\hat{J}_j(1) - \hat{K}_j(1)], \quad (5.28)$$

Where \hat{H}^{core} is the hydrogenic (core) hamiltonian

$$\hat{H}^{\text{core}}(1) = -\frac{1}{2}\nabla_1^2 - \frac{Z}{r_1}, \quad (5.29)$$

and \hat{J}_j is the Coulomb operator and \hat{K}_j is the exchange operator. The matrix elements of the Fock operator have the form

$$F_{pq} = \langle \phi_p(1) | \hat{H}^{\text{core}}(1) | \phi_q(1) \rangle + \sum_{j=1}^n [2\langle \phi_p(1) | \hat{J}_j(1) | \phi_q(1) \rangle - \langle \phi_p(1) | \hat{K}_j(1) | \phi_q(1) \rangle] \quad (5.30)$$

This can be rewritten for simplicity as

$$F_{pq} = H_{pq}^{\text{core}}(1) + G_{pq}, \quad (5.31)$$

where

$$G_{pq} = \sum_{u,v=0}^{m-1} D_{u,v} [2(pq|uv) - (pv|uq)]. \quad (5.32)$$

$D_{u,v} = \sum_{j=1}^n c_{uj}^* c_{vj}$ represents the density matrix, $(pq|uv)$ the Coulomb integral J and $(pv|uq)$ the exchange integral K . $(pq|uv)$ is a compact way of writing a two electron integral which has the form

$$(pq|uv) = \int \int \phi_p^*(1) \phi_q(1) \frac{1}{r_{12}} \phi_u^*(2) \phi_v(2) d\nu_1 d\nu_2. \quad (5.33)$$

Consider the energy of an electron, ϵ_i , given by the Fock operator in equation (5.28)

$$\epsilon_i = \int \psi \hat{F} \psi dv = H_{ii}^{core} + \sum_{j=1}^n (2J_{ij}(1) - K_{ij}(1)). \quad (5.34)$$

If all the one-electron energies are summed together this in turn doubles the effect of the electron-electron interaction. When calculating the full electronic energy, the energy calculated by the Fock operator given in equation (5.34) is doubled and the double-counted superfluous electron-electron repulsion energy, E_{ee} , is removed. The HF energy is thus calculated as

$$E_{HF} = \underbrace{2 \sum_{i=1}^n \epsilon_i}_{2\langle \hat{F} \rangle} - \underbrace{\sum_{i=1}^n \sum_{j=1}^n (2J_{ij}(1) - K_{ij}(1))}_{E_{ee}}. \quad (5.35)$$

In this work only singlet states are calculated, so $2J - K$ reduces to J as the system is closed-shell (no parallel spins, thus $K = 0$). Therefore the HF energy is calculated as

$$E_{HF} = 2\epsilon_i - J_{ij}, \quad \text{where } i, j = 1 \text{ or } 2, i \neq j. \quad (5.36)$$

5.2.2.1 Implementation

To ensure accuracy of results, the main HF Map1e program that calculates energies and wavefunctions was set-up to use quadruple precision (32 digits), with the C++ code that calculates the two-electron integrals running at arbitrary precision due to large factorials which require 100+ digits of accuracy. For comparative purposes the fully correlated calculations were also run at quadruple precision.

One electron integrals: Series solution

As Laguerre polynomials were selected for the basis, the series solution was found to

work in the calculation of the one electron integrals. Just as before, the Laguerre polynomial series solution is possible as the coordinates range from zero to infinity and the Laguerre orthogonality relationship is satisfied

$$\int_0^{\infty} e^{-x} L_p(x) L_q(x) dx = \delta_{p,q}. \quad (5.37)$$

This series solution was programmed using Maple in a similar manner to the FC method in Chapter 3. The wavefunction given in equation (5.27) is substituted into

$$(\hat{H}^{core}(1) - \epsilon^{core})\psi = 0. \quad (5.38)$$

The Laguerre recurrence relations eliminate the derivatives and powers of the variable r_i , arising from the one-electron operators. This process leads to a 5-term recursion relation, between the coefficients $C(q)$ of the form^[153]

$$\sum_{\alpha=-2}^{+2} R_{\alpha}(q) C(q + \alpha) = 0. \quad (5.39)$$

5-terms are generated due to the application of the Laguerre recurrence relations to the Laplacian in the kinetic energy operator, resulting in 5 unique terms, $\{q - 2, q - 1, q, q + 1, q + 2\}$. Table 5.1 lists the coefficients for the general case where Z and A are arbitrary.

Table 5.1 Coefficients $R_{\alpha}(q)$ in the recursion relation given by Eq. 5.39 for the nuclear charge Z and variational parameter A .

α	$R_{\alpha}(q)$
-2	$-q\pi(A^2q - A^2 + 8q - 8)/2A^3$
-1	$4q\pi(AZ + 4q)/A^3$
0	$\pi(A^2q^2 + A^2q - 8AZq + A^2 - 4AZ - 24q^2 - 24q - 8)/A^3$
+1	$4\pi(AZq + AZ + 4q^2 + 8q + 4)/A^3$
+2	$-\pi(A^2q^2 + 3A^2q + 2A^2 + 8q^2 + 24q + 16)/2A^3$

The recursion relation only needs to be calculated once, and can then be used to calculate any of the m^2 matrix elements. An example of a 3×3 determinant is given in equation (5.40) where A and Z are arbitrary.

$$\det(H^{core} - \epsilon^{core} S) = \begin{vmatrix} \frac{\pi}{A} - \frac{4\pi Z}{A^2} - \epsilon^{core} \frac{8\pi}{A^3} & \frac{4\pi Z}{A^2} + \epsilon^{core} \frac{16\pi}{A^3} & -\frac{\pi}{A} - \epsilon^{core} \frac{8\pi}{A^3} \\ \frac{4\pi Z}{A^2} + \epsilon^{core} \frac{16\pi}{A^3} & \frac{3\pi}{A} - \frac{12\pi Z}{A^2} - \epsilon^{core} \frac{56\pi}{A^3} & \frac{8\pi Z}{A^2} + \epsilon^{core} \frac{64\pi}{A^3} \\ -\frac{\pi}{A} - \epsilon^{core} \frac{8\pi}{A^3} & \frac{8\pi Z}{A^2} + \epsilon^{core} \frac{64\pi}{A^3} & \frac{7\pi}{A} - \frac{20\pi Z}{A^2} - \epsilon^{core} \frac{152\pi}{A^3} \end{vmatrix} \quad (5.40)$$

Evaluating this determinant gives its following general form for a 3 term HF wavefunction

$$\begin{aligned} \det(H^{core} - \epsilon^{core} S) = \frac{6\pi^3}{A^9} & \left(3A^6 - 36A^5Z + 96A^4Z^2 - 64A^3Z^3 \right. \\ & - 168A^4(\epsilon^{core}) + 832A^3Z(\epsilon^{core}) - \\ & 768A^2Z^2(\epsilon^{core}) + 1344A^2(\epsilon^{core})^2 \\ & \left. - 2304ZA(\epsilon^{core})^2 - 1536(\epsilon^{core})^3 \right). \end{aligned} \quad (5.41)$$

Solving the one electron integrals via a recursion relation was found to be approximately 1000 times faster than explicit integration when using 20 terms in the wavefunction.

Two electron integrals: Analytical form

The series solution method used for the calculation of the one-electron integrals cannot be applied to the two electron integrals, as terms arise which do not satisfy the Laguerre orthogonality condition seen in equation (5.37). The two electron integrals have the following general form

$$\int_0^\infty \int_0^\infty \int_{|r_1-r_2|}^{r_1+r_2} e^{-r_1-r_2} L_p(r_1) L_q(r_1) \frac{1}{r_{12}} L_u(r_2) L_v(r_2) dr_{12} dr_2 dr_1. \quad (5.42)$$

A full derivation of an analytical solution to this integral is presented in Appendix C, which complements the derivation provided by co-investigator A. W. King^[77]. This direct derivation was required for writing a Maple script to demonstrate the accuracy of the final formula used in the HF implementation. It involves firstly transforming from inter-particle to perimetric coordinates, and writing the Laguerre polynomials in the form

$$L_n(x) = \sum_{n_i=0}^n (-1)^{n_i} \binom{n}{n_i} \frac{x^{n_i}}{n_i!}. \quad (5.43)$$

By expanding the binomial expansions, terms are left which are compatible with the known integral^[55]

$$\int_0^\infty x^n e^{-ax} dx = \Gamma(n+1) a^{-n-1}. \quad (5.44)$$

This leads to an analytical expression for the two-electron integrals with the one stated in equation (5.42) having the form

$$\begin{aligned} & \sum_{p_i, q_i, u_i, v_i=0}^{p, q, u, v} \sum_{a_i=0}^{p_i+q_i} \sum_{b_i=0}^{u_i+v_i} 2^{a_i+b_i-\phi+1} (-1)^{\phi+1} \pi^2 \left(a_i^2 - 2a_i b_i + b_i^2 - p^2 - 2p_i q_i - 2p_i u_i - 2p_i v_i \right. \\ & \quad \left. - q_i^2 - 2q_i u_i - 2q_i v_i - u^2 - 2u_i v_i - v^2 + a_i + b_i - 7\phi - 10 \right) (\phi - a_i - b_i)! (p_i + q_i) \\ & \quad \times \frac{(u_i + v_i)! p! q! u! v!}{u_i!^2 v_i!^2 u_i!^2 v_i!^2 (p_i + q_i - a_i)! (q_i + u_i - b_i)! (p - p_i)! (q - q_i)! (u - u_i)! (v - v_i)!} \end{aligned} \quad (5.45)$$

Where $\phi = p_i + q_i + u_i + v_i$.

Computational bottleneck

The calculation of the two-electron integrals, m , is the main computational bottleneck in this HF implementation. Overall there are a minimum of m^4 two-electron integrals, resulting in poor scaling when more terms are included in the wavefunction. A lot of these integrals will be repeated such as $(pq|uv) = (pq|vu)$ or $(pq|uv) = (uv|pq)$, so only unique two-electron integrals are calculated by exploiting the 8-fold permutation symmetry of the problem.

- Four unique indices, $p \neq q \neq u \neq v \implies 3 \binom{m}{4}$
- Three unique indices, e.g. $p \neq q \neq u = v \implies 6 \binom{m}{3}$
- Two unique indices, e.g. $p \neq q = u = v \implies 4 \binom{m}{2}$
- Zero unique indices, $p = q = u = v \implies 1 \binom{m}{1}$

These add to give a total of $m(m+1)(m^2+m+2)/8$ two-electron integrals. For a wavefunction with 20-terms this permutation symmetry significantly reduces the total number of two-electron integrals that need calculating from 160,000 to 22,155.

A specialised C++ code was implemented by A. W. King devoted to calculating these two-electron integrals as it was found Maple was too slow. This code is simple and calculates the two-electron integrals using the form given in equation (C.18)

as a quadruple for loop. This analytical form contains a large number of factorial expressions which require a very high level of precision as the number of terms in the wavefunction increases. For a 20-term wavefunction factorials arise requiring over 100 digits of accuracy to be carried through the calculation. Originally this code used the `mpfr` library, but the current author implemented the `gmp` library via operator overloading², as it was found to perform better than `mpfr` for big number arithmetic.

To calculate every unique two-electron integral for a 20-term wavefunction all unique combinations of $(pq|uv)$ less than 20 are iterated over. When calculating the integral $(20\ 20|20\ 20)$ nearly all of the lower integrals need to be performed as well so this is exploited in the code by calculating the largest integral first and saving the intermediary steps to memory which can be extracted and re-used. This reduces the computational expense whilst simultaneously causing a substantial increase in the required RAM, which currently limits the wavefunction size to 20 terms on the available hardware.

Future optimisations are warranted such as optimising how the two-electron integrals are assigned to the matrix, G_{pq} given in equation (5.32), which is currently both a time and memory consuming process done by Maple. The memory requirements could be dropped substantially by memory mapping the matrix elements from file, where a portion of virtual memory would be assigned a direct byte-to-byte correlation with the file containing the matrix elements in order to treat it as though it were in primary memory. This would require a substantial rewrite in C++, Python or Fortran as it is beyond the scope of the Maple language, but should allow larger wavefunctions to be used providing even more accurate results.

The C++ and Python languages contain a powerful programming idiom called Resource Acquisition Is Initialisation (RAII)^[69], sometimes called Scope Bound Resource Management (SBRM) which uses constructors/destructors to very efficiently manage memory, and it is possible to know when memory becomes free to efficiently reallocate it. This would require careful planning but would improve the efficiency of memory management in calculation of the two-electron integrals which is the bottleneck for the HF code.

Calculation sequence

There are many steps involved in the HF method so a summary is given below. The HF code development and following calculation steps were inspired by the insightful

²Where a function can have multiple definitions.

tutorial of Dr. Yukio Yamaguchi^[156].

Hartree-Fock calculation sequence

- Step 1.** Define the charge and wavefunction for the system under consideration.
- Step 2.** Calculate one electron integrals using recursion relation and assign these to matrices using Maple code.
- Step 3.** Diagonalise the one electron integrals as the secular determinant $|\mathbf{H} - E\mathbf{S}| = 0$. At this stage the matrix does not have electron-electron interactions and therefore does not require the Self Consistent Field (SCF) process. The coefficients from this diagonalisation become the guess coefficients for the steps to follow.
- Step 4.** The initial guess coefficients are used to create the two electron integral matrix.
- Step 5.** Create Fock matrix as the sum of the one electron and two electron matrices.
- Step 6.** Diagonalise the Fock matrix as the secular determinant $|\mathbf{F}| = 0$ and obtain new coefficients.
- Step 7.** Use new coefficients to calculate the two electron integral matrix.
- Step 8.** Repeat the previous three steps until the coefficients stop changing, which means the Self Consistent Field procedure has converged.
- Step 9.** Calculate the Hartree-Fock energy as equation (5.35).
- Step 10.** Change NLP A and return to step 2, until these parameters converge to a required tolerance (in this instance the tolerance is set to 3×10^{-15}). This is optimised via Maple's inbuilt `NLPSolve` command, which by default has been set to use quadratic interpolation; but other methods can be specified by the user.

5.2.3 Coulomb hole calculation

The Coulomb hole was defined by Coulson and Nielson^[141] as the difference in the distribution function of the inter-electronic distance r_{12} for the correlated wavefunction and the Hartree-Fock wavefunction. The difference in the electron density calculated using the FC and HF methods, provides a measure of the effects of electron correlation on the electron density itself. One of these effects is referred to as the Coulomb hole caused by the instantaneous repulsive interaction between electrons^[142,157–159]; which acts to decrease the probability of finding electrons near one another. This

instantaneous effect is not seen in the HF method due to electron correlation being modelled using a mean field approximation. The Coulomb hole quantifies the error from HF using a mean-field approach.

In order to calculate the Coulomb hole, the radial distribution function along the r_{12} coordinate is calculated for both the FC and HF methods. The radial distribution function is calculated using

$$D(r) = 4\pi r^2 \langle \delta(r_{12} - r) \rangle, \quad (5.46)$$

where the computational techniques are discussed in Chapter 3. The difference between the intracule distribution functions generated with the statistically independent and uncorrelated HF approximation and the explicitly correlated FC method defines the Coulomb hole, i.e.

$$\Delta D(r) = D_{FC}(r) - D_{HF}(r), \quad (5.47)$$

where $\Delta D(r)$ is calculated as the numerical difference between $D_{FC}(r)$ and $D_{HF}(r)$. To address the high level of accuracy that is required with this method a high accuracy C++ code was written which is able to calculate particle densities to 15-16 decimal places of accuracy.

5.2.3.1 Implementation

Calculating the intracule distribution function given in equation (5.46) can be done for both the FC and HF wavefunctions in Maple, but in order to achieve the very high accuracy required for the Coulomb holes a more robust method and lower level language is used, namely C++. Whilst Maple can calculate these intracule densities, its accuracy at higher r was found to be prone to numerical rounding errors which is undesirable when looking for small Coulomb holes. A second Coulomb hole was correctly identified but a tertiary Coulomb hole was incorrectly produced when integrating in Maple due to floating point rounding errors; which does not appear when using the Cuhre numerical integration technique discussed in Chapter 3. A benefit of using this numerical method (Cuhre) is that convergence of each calculated data point is done by the algorithm to confirm the validity of the result.

In order to use the CUBA numerical integration library containing Cuhre, the integration domains must be made integrable over a unit hypercube as discussed in Chapter 3. The HF wavefunction is in terms of r_1 and r_2 only and in order to plot the spherically averaged HF intracule density (in inter-particle coordinates)

$$D(r) = 4\pi r^2 \int_0^\infty \int_0^\infty \int_{|r_1-r_2|}^{|r_1+r_2|} \Psi(r_1, r_2, r_{12}) \delta(r_{12} - r) \Psi(r_1, r_2, r_{12}) dr_{12} dr_2 dr_1, \quad (5.48)$$

the r_{12} coordinate is required. The inter-particle coordinates are transformed into perimetric coordinates, $z_i = r_j + r_k - r_i$ via the following transformations (Jacobian factors are correctly accounted for in the code but not written here)

$$\begin{aligned} r_1 &= \frac{z_2 + z_3}{2} \\ r_2 &= \frac{z_3 + z_1}{2} \\ r_{12} &= \frac{z_1 + z_2}{2} \end{aligned} \quad (5.49)$$

to give

$$D(r) = 4\pi r^2 \int_0^\infty \int_0^\infty \int_0^\infty \Psi(z_1, z_2, z_3) \delta\left(\frac{z_1 + z_2}{2} - r\right) \Psi(z_1, z_2, z_3) dz_3 dz_2 dz_1. \quad (5.50)$$

z_2 can be eliminated by the making the following substitution

$$r_{12} = \frac{z_1 + z_2}{2} \rightarrow 2r_{12} = z_1 + z_2 \quad \therefore z_2 = 2r_{12} - z_1. \quad (5.51)$$

This substitution eliminates z_2 and also sets the integration limit for z_1 as z_2 cannot be negative and neither can z_1 . The maximum value z_1 can take is $z_1 = 2r_{12}$ ($z_2 = 0$) and the minimum value $z_1 = 0$. After integrating the Dirac delta distribution, the intracule density can be calculated using the following form with independent integration domains

$$D(r) = 4\pi r^2 \int_0^\infty \int_0^{2r} \Psi(z_1, z_3) \Psi(z_1, z_3) dz_1 dz_3. \quad (5.52)$$

A coordinate mapping is then done to transform the integration domains to $0 \rightarrow 1$ with the use of a dummy variable. The integrand is simplified and saved in a C++ recognisable format which is loaded and processed by the C++ code as described in Chapter 3. By default the step size for the density plot is set to 0.005, which means over the range $0 \rightarrow 20$ a.u. 4000 data points are calculated ensuring a smooth accurate density plot.

Coulomb hole properties

In order to calculate properties of the Coulomb holes including roots, area and minimum the data points of $\Delta D(r)$, Eq. (5.47), were fitted to a spline curve, $f(a)$, of the form

$$f(a) = \begin{cases} C_{0,0} + C_{1,0}a + C_{2,0}a^2 + C_{3,0}a^3 & a < R_0 \\ C_{0,1} + C_{1,1}a + C_{2,1}a^2 + C_{3,1}a^3 & a < R_1 \\ \vdots & \vdots \\ C_{0,n-1} + C_{1,n-1}a + C_{2,n-1}a^2 + C_{3,n-1}a^3 & a < R_{n-1} \\ C_{0,n} + C_{1,n}a + C_{2,n}a^2 + C_{3,n}a^3 & \text{otherwise,} \end{cases} \quad (5.53)$$

where n is the number of numerical data points used to form the $D(r)$ curves which in this instance is 4000. The R_i values are the r values for each of these data points. The $C_{i,j}$ are determined by the cubic spline fitting algorithm built into Maple, which was confirmed by two separate spline fitting algorithms written in C++ and Python. The spline curve is a function and thus differentiation can be used to find stationary points along with integration to calculate the area.

5.3 Results and Discussion

5.3.1 Literature comparison

Table (5.2) compares the energy results calculated for the helium isoelectronic series using this new HF implementation against the most accurate literature values.

Table 5.2 Calculated HF energies compared to literature.

Species	Hartree Fock Energy (E_h)		Ref.
	This Work	Literature	
H ⁻	-0.487 929 734 369 2	-0.487 929 734 372	[160]
He	-2.861 679 995 612	-2.861 679 995 612	[161]
Li ⁺	-7.236 415 201 452	-7.236 415 201 431 42	[162]
Be ²⁺	-13.611 299 430 62	-13.611 299 430 604 0	[163]
B ³⁺	-21.986 234 466 82	-21.986 234 466 814 1	[163]
C ⁴⁺	-32.361 192 875 72	-32.361 192 875 710 1	[163]
N ⁵⁺	-44.736 163 964 94	-44.736 163 964 935 6	[163]
O ⁶⁺	-59.111 142 701 91	-59.111 142 701 907 8	[163]
F ⁷⁺	-75.486 126 406 26	-75.486 126 406 255 1	[163]
Ne ⁸⁺	-93.861 113 519 23	-93.861 113 519 195 7	[163]
Na ⁹⁺	-114.236 103 072 5	-114.236 103 1	[164]
Mg ¹⁰⁺	-136.611 094 432 9	-136.611 094 4	[164]
Al ¹¹⁺	-160.986 087 168 8	-160.986 087 2	[164]
Si ¹²⁺	-187.361 080 975 9	-187.361 081 0	[164]
P ¹³⁺	-215.736 075 633 6	-215.736 075 6	[164]
S ¹⁴⁺	-246.111 070 977 8	-246.111 071 0	[164]
Cl ¹⁵⁺	-278.486 066 884 2	-278.486 066 9	[164]
Ar ¹⁶⁺	-312.861 063 256 8	-312.861 063 3	[164]

The energies are in excellent agreement with the most accurate literature values and offer increased accuracy for all of the cations.

5.3.2 Energy Convergence

Table 5.3 lists energy convergence data for helium calculated using the aforementioned HF implementation with an increasing number of terms used in the wavefunction.

Table 5.3 The rate of convergence of the helium HF energy (hartree) with number of basis functions m . Digits in bold represent converged digits with respect to the numerical Hartree-Fock value^[161] and digits in bold-italic represent increased convergence exhibited by the method in this work in agreement with^[165]. Also provided is the helium core energy, $2 \times \epsilon^{core}$, and the optimised value of the NLP A .

m	A	HF Energy / a.u.	$2 \times \epsilon^{core}$ / a.u.
1	3.374 999	-2.847 656 249 999 999	-3.902 343 749 999 996
3	3.841 765	-2.861 590 054 663 477	-3.999 999 993 527 671 ^a
5	3.933 089	-2.861 679 675 254 110	-3.999 999 999 999 999
8	4.683 471	-2.861 679 963 194 483	-3.999 999 999 999 999
10	4.997 241	-2.861 679 993 922 506	-3.999 999 999 999 999
12	5.217 990	-2.861 679 995 536 817	-3.999 999 999 999 999
14	5.383 991	-2.861 679 995 609 217	-3.999 999 999 999 999
16	5.506 436	-2.861 679 995 612 136	-3.999 999 999 999 999
18	5.581 394	-2.861 679 995 612 237	-3.999 999 999 999 999
20	5.581 838	-2.861 679 995 612 239	-3.999 999 999 999 999
Literature/Exact	N/A	-2.861 679 995 612 2 ^[161]	-4

^a Verifiable by substituting, $Z = 2$ and $A = 3.841\,765$ into equation (5.41).

With only three terms in the wavefunction, the method accurately calculates the helium core energy, $2\epsilon^{core}$, to within a tenth of a μ -hartree. At 14 terms the HF energy matches the exact numerical HF value^[166] and at 18 terms agrees with the value from Gázquez^[165]. The high accuracy of these results provides confidence that the HF method is working as expected. Inclusion of the NLP A offers a substantial advantage for energy convergence highlighted in Table 5.4.

Table 5.4 Effect of the NLP A on the energy of a 10- and 20-term HF wavefunction for helium. Digits in bold represent converged digits with respect to the numerical Hartree-Fock value^[166].

A	Number of basis functions, m	
	10	20
1	-2.852 188 594 931	-2.861 643 004 528
2	-2.861 664 390 672	-2.861 679 942 413
Opt*	-2.861 679 993 922	-2.861 679 995 612
Numerical HF ^[166]	-2.861 679 995 6	

* $A = 4.997\,241\,294$ for $m = 10$ and $A = 5.581\,838\,558$ for $m = 20$.

When $A = 1$ this is equivalent to not including the parameter, and with a value of $A = Z = 2$ correctly matches the asymptotic behaviour for a two-electron atom

with $Z = 2$ at large r ($A = \sqrt{-E}$,^[7]). The 10-term wavefunction gains an additional 4 converged digits and the 20-term wavefunction an extra 5-6 converged digits when relaxing any constraint on the value of A , allowing for increased efficiency by fewer terms being needed in the wavefunction.

5.3.3 Wavefunction Quality: Bound State Properties

With the energies achieving very high accuracy the quality of the wavefunctions is now assessed along with the effects of correlation on other key properties, Table 5.5 provides FC and HF expectation values $\langle \hat{X} \rangle$ where

$$\hat{X} = \langle \psi_{\text{FC}} | \hat{X} | \psi_{\text{FC}} \rangle \text{ or } \langle \psi_{\text{HF}} | \hat{X} | \psi_{\text{HF}} \rangle. \quad (5.54)$$

The properties presented include the expectation value of the interparticle distances r_i , the two-particle Dirac delta functions, $\delta(r_i)$, virial condition η and the two-particle cusps, ν_{ij} . The calculated values for η in Table 5.5 are less than 6.2×10^{-20} for the FC systems and 5.4×10^{-14} for the HF systems, which are close to the exact value of zero.

Included in Table 5.5 are results for the FC and HF Z_C systems, where for the FC method, the critical nuclear charge to bind two electrons to the nucleus is, $Z_C^{FC} = 0.911\,028\,224\,077\,255\,73$ ^[47]. The critical nuclear charge to bind two electrons to the nucleus under the HF regime is calculated later in this chapter for the first time.

Table 5.5 Expectation values (in a.u.), cusps and virial condition for Z_C , H^- , He and Li^+ using either the fully-correlated (FC)^a or the Hartree-Fock (HF) wavefunction. The exact value of the nucleus-electron cusp $\nu_{31} = -Z$, electron-electron cusp $\nu_{12} = 0.5$, and the virial condition $\eta = 0$.

$\langle \hat{X} \rangle$	Z_C		H^-	
	FC	HF	FC	HF
$\langle r_1 \rangle$	4.146	2.989 00	2.710 178 27	2.503 959
$\langle r_{12} \rangle$	7.083	4.493 9	4.412 694 4	3.739 273
$\langle \delta(r_1) \rangle$	0.119 094	0.108 4	0.164 552 8	0.154
$\langle \delta(r_{12}) \rangle$	0.001 114	0.008 601 2	0.002 738 0	0.012 983 476 3
$\langle 1/r_{12} \rangle$	0.223 374	0.337 767	0.311 021 502 2	0.395 484 8
ν_{31}	-0.910 981	-0.911 190	-0.999 999 98	-1.000 047
ν_{12}	0.498 685	n/a	0.499 946	n/a
η	6.13×10^{-20}	2.20×10^{-15}	2.87×10^{-21}	5.34×10^{-14}
	He		Li^+	
	FC	HF	FC	HF
$\langle r_1 \rangle$	0.929 472 294 87	0.927 273 404 7	0.572 774 149 971	0.572 366 815 001
$\langle r_{12} \rangle$	1.422 070 255 56	1.362 124 383 6	0.862 315 375 45	0.838 314 780 311
$\langle \delta(r_1) \rangle$	1.810 429	1.797 959	6.852 009	6.836 07
$\langle \delta(r_{12}) \rangle$	0.106 345 37	0.190 603 997 806	0.533 722	0.770 240 340 922
$\langle 1/r_{12} \rangle$	0.945 818 448	1.025 768 869	1.567 719 559 13	1.651 686 396 96
ν_{31}	-1.999 999 91	-1.999 999 4	-2.999 999 92	-2.999 999 2
ν_{12}	0.499 993	n/a	0.499 996	n/a
η	8.70×10^{-22}	4.53×10^{-17}	5.78×10^{-23}	7.12×10^{-19}

^a FC values reported agree with^[167] for H^- ,^[39] for He, and^[38] for Li^+ .

The exact value of ν_{31} , the nucleus-electron cusp, is $-\frac{ZM}{1+M}$ which reduces to $-Z$ for the infinite nuclear mass systems considered here, and the exact value of ν_{12} , the electron-electron cusp is 0.5, but is zero for Hartree-Fock calculations^[163].

The nucleus-electron (single-particle) probability density $\rho(r)$ can be calculated using $\langle \delta(r_i - r) \rangle$, $i = 1$ or 2 and by setting $r = 0$ the electronic structure near the nucleus can be elucidated. Similarly the intracule (electron-pair) density can be calculated using $\langle \delta(r_{12} - r) \rangle$. These are represented as $\langle \delta(r_1) \rangle$ and $\langle \delta(r_{12}) \rangle$ in Table (5.5) and physically represent the probability of r_1 or r_{12} being equal to zero, i.e. the coalescence probability of the two particles.

The effect of electron correlation on $\langle \delta(r_1) \rangle$ is less than 1 % for all but the anions, H^- and Z_C , where the error introduced by the Hartree-Fock approximation is more substantial. The correlation effects are significantly larger for $\langle \delta(r_{12}) \rangle$ for all systems, and although the % error decreases as the nuclear charge increases caused by the dominant nucleus-electron interaction^[95], it is still significant. This leads to the

conclusion that HF provides a reasonable description for the single-particle density but not for the electron-pair density in agreement with previous work^[163].

The virial condition, cusp condition and accurate expectation values, in addition to the good precision for the energies, indicate that a reasonable description for the HF wavefunctions have been obtained and that the new HF implementation works as expected. These HF wavefunctions have been made available via the Supplementary information of the HF paper published on this work^[153].

5.3.4 Electron Correlation Energies

The Löwdin definition of electron correlation given in equation (5.1) is the difference in the “exact” energy and the HF energy. In this work the “exact” energy is assumed to be the Fully Correlated (FC), non-relativistic three-body energy with an infinite nuclear mass. An infinite nuclear mass is required in order for appropriate comparison between the FC and HF methodologies where the implemented Laguerre HF method is only currently usable within the clamped nucleus approximation, where extensions for finite nuclear mass make for possible future work. For all energy calculations the matrix size used in the FC method is 4389 and for the HF is 20. Table 5.6 contains HF and FC data for the helium isoelectronic series up to Ar^{16+} , along with the difference between them (E_{corr}).

Table 5.6 The Hartree-Fock energy E_{HF} , fully-correlated energy E_{FC} , and the correlation energy $E_{corr} = E_{FC} - E_{HF}$, accurate to the number of digits presented. All energies are given in a.u.

System	E_{HF}	E_{FC}	E_{corr}
H ⁻	-0.487 929 734 3	-0.527 751 016 54	-0.039 821 282
He	-2.861 679 995 612	-2.903 724 377 034 0	-0.042 044 381 422
Li ⁺	-7.236 415 201 452 2	-7.279 913 412 66	-0.043 498 211 20
Be ²⁺	-13.611 299 430 619 1	-13.655 566 238 42	-0.044 266 807 80
B ³⁺	-21.986 234 466 824 34	-22.030 971 580 24	-0.044 737 113 41
C ⁴⁺	-32.361 192 875 718 01	-32.406 246 601 89	-0.045 053 726 17
N ⁵⁺	-44.736 163 964 942 47	-44.781 445 148 7	-0.045 281 183 7
O ⁶⁺	-59.111 142 701 913 658	-59.156 595 122 7	-0.045 452 420 7
F ⁷⁺	-75.486 126 406 260 515	-75.531 712 363 9	-0.045 585 957 6
Ne ⁸⁺	-93.861 113 519 231 344	-93.906 806 515 0	-0.045 692 995 7
Na ⁹⁺	-114.236 103 072 533 87	-114.281 883 776 0	-0.045 780 703 4
Mg ¹⁰⁺	-136.611 094 432 922 60	-136.656 948 312	-0.045 853 879 0
Al ¹¹⁺	-160.986 087 168 814 85	-161.032 003 026 0	-0.045 915 857 1
Si ¹²⁺	-187.361 080 975 930 946	-187.407 049 998 6	-0.045 969 022 6
P ¹³⁺	-215.736 075 633 578 38	-215.782 090 763 5	-0.046 015 129 9
S ¹⁴⁺	-246.111 070 977 793 26	-246.157 126 474	-0.046 055 496
Cl ¹⁵⁺	-278.486 066 884 212 90	-278.532 158 015	-0.046 091 130
Ar ¹⁶⁺	-312.861 063 256 800 63	-312.907 186 076 5	-0.046 122 819 6

The correlation energies in Table 5.6 are in excellent agreement with literature values (^[168] for H⁻, He, Li⁺, B³⁺, Ne⁸⁺ and ^[169] for Be²⁺).

5.3.5 Critical nuclear charge for binding under the Hartree-Fock approximation

One of the benefits of implementing our own Hartree-Fock method is complete control over the optimisation of the wavefunction, including for arbitrary nuclear charge Z . The critical nuclear charge, Z_C is the minimum charge required for an atomic system to have at least one bound state. Previous work within the research group^[170] and by others^[47,171] has been dedicated to finding this charge value for fully correlated systems with the most accurate value calculated as $Z_C^{\text{FC}} = 0.911\,028\,224\,077\,255\,73$ ^[47]. However no such value has been determined for a two-electron atom described by the HF method. Even before calculating Z_C for HF, we know that it will be

higher than this FC value as HF predicts the hydride ion to be unbound, as seen in Table 5.6.

A variational approach was used to determine $Z_C^{\text{FC}[94]}$ which unfortunately is not possible for the HF method due to the SCF procedure. A more direct approach is taken and a minimisation procedure was written in Maple with the help of the `fsolve` command; which varies Z until E_{HF} equals the lowest continuum threshold which in this instance is the hydrogenic energy, $E = -Z^2/2$ a.u. This resulting critical charge value is $Z_C^{\text{HF}} = 1.031\,177\,528$ with convergence data shown in Table 5.7.

Table 5.7 Convergence of the Hartree Fock critical nuclear charge value, Z_C^{HF} .

m	Z_C^{HF}
2	1.066 941 738 12
4	1.032 148 574 04
6	1.031 203 748 30
8	1.031 178 034 36
10	1.031 177 599 42
12	1.031 177 539 47
14	1.031 177 529 68
16	1.031 177 528 42
18	1.031 177 528 31
20	1.031 177 528 26

Correlation energy at the critical nuclear charge

Table 5.6 shows that the HF method produces very accurate energy and correlation energy results for the helium isoelectronic series which gives confidence when investigating weakly bound systems such as those that have the critical nuclear charge. Table 5.8 shows the correlation energies for $Z_C^{\text{HF}} = 1.031\,177\,528$ calculated using a 20 term wavefunction as described in the previous section and in the publication^[153], and $Z_C^{\text{FC}} = 0.911\,028\,224\,077\,255\,73$ ^[47] representing the critical nuclear charge for binding two-electrons in a FC system with infinite nuclear mass, calculated using a 4389 term wavefunction.

Table 5.8 The Hartree-Fock energy E_{HF} , fully-correlated energy E_{FC} , and the correlation energy $E_{corr} = E_{FC} - E_{HF}$ for the HF critical nuclear charge $Z_C^{HF}=1.031\,177\,528$ and FC critical nuclear charge $Z_C^{FC} = 0.911\,028\,224\,077\,255\,73^{[47]}$, accurate to the number of digits presented. All energies are given in a.u.

Z	E_{HF}	E_{FC}	E_{corr}
$Z_C^{HF}=1.031\,177\,528$	-0.531 663 54	-0.571 378 664 40	-0.039 715 124 44
$Z_C^{FC} = 0.911\,028\,224$	-0.373 906 22	-0.414 986 21	-0.041 079 99

The critical nuclear charge is higher for HF theory due to it underestimating the energy of the system caused by a lack of dynamic electron correlation; thus a greater nuclear charge is required to form a bound state. Interestingly, the magnitude of the correlation energy for Z_C^{FC} is higher than that for H^- .

Low nuclear charge systems

Highly accurate correlation energies and HF wavefunctions have been calculated for the helium isoelectronic sequence. For systems where $Z < Z_C^{HF}$ the ground state is unstable to electron detachment. Calculating the HF wave function accurately for any nuclear charge, including low, non-integer values is beneficial for method testing and to understand the long-range behaviour of electron correlation. The robustness of the implemented HF method along with the behaviour below Z_C^{HF} is explored by varying Z below Z_C^{HF} and analysing the convergence in energy and the expectation values of r_1 and r_{12} . Due to the HF wavefunctions having a maximum number of 20 terms, the expectation values are calculated using Maple by means of the explicit integration

$$\langle r_1 \rangle = \int \Psi r_1 \Psi dV, \quad \langle r_{12} \rangle = \int \Psi r_{12} \Psi dV. \quad (5.55)$$

Expectation values, $\langle r_1 \rangle, \langle r_{12} \rangle$ are listed in Table 5.9 for nuclear charge values of 2, Z_C^{HF} , 1, Z_C^{FC} , 0.9, 0.85 and 0.8.

Table 5.9 Expectation value convergence for low, nuclear charge Z as a function of basis set size m . Bold digits represent converged digits, and helium is included for comparison along with the threshold energy E_{th} for each system. Radial expectation values are in a.u.

Z	E_{th}	m	$\langle r_1 \rangle$	$\langle r_{12} \rangle$
2	-2	12	0.927 273 403 133	1.362 124 380 627
		16	0.927 273 404 727	1.362 124 383 668
		20	0.927 273 404 731	1.362 124 383 676
Z_C^{HF} =1.031 177 528	-0.531 663 547	12	2.372 690 028	3.537 391 966
		16	2.372 691 771	3.537 395 323
		20	2.372 691 816	3.537 395 411
1	-0.5	12	2.503 956 640	3.739 268 219
		16	2.503 959 062	3.739 272 889
		20	2.503 959 636	3.739 273 998
Z_C^{FC} = 0.911 028 224	-0.414 986 212	12	2.988 986 931	4.493 907 504
		16	2.989 002 458	4.493 937 611
		20	2.989 003 842	4.493 940 297
0.9	-0.405	12	3.065 294 750	4.614 076 439
		16	3.065 313 178	4.614 112 237
		20	3.065 315 315	4.614 116 388
0.85	-0.361 25	12	3.487 529 820	5.288 177 405
		16	3.487 705 173	5.288 517 822
		20	3.487 732 757	5.288 571 638
0.80	-0.32	12	4.213 237 569	6.507 070 336
		16	5.130 527 896	8.313 287 004
		20	6.474 496 685	10.968 420 616

As was seen for the energies in Table 5.10, lowering the nuclear charge to $Z = 0.85$ results in acceptable convergence, which for the radial expectation values is 4-5 significant figures. The convergence deteriorates rapidly as the nuclear charge is decreased as for the energies, where for $Z = 0.8$ the radial expectation values have 0 converged digits. For $Z = 0.8$, $\langle r_1 \rangle$ and $\langle r_{12} \rangle$ are seen to increase suddenly, doubling in the case of $\langle r_{12} \rangle$ confirming that results calculated with this HF method are invalid for $Z < 0.85$.

More basis functions may be needed to accurately model systems with $Z < 0.8$ or the wavefunction may be inadequate at capturing the physical behaviour for systems where $Z < 0.85$ and more investigation is needed into the variational wave function at and below the critical point making for important future work.

When the nuclear charge reaches the critical charge value it would be expected that one of the electrons moves to an infinite separation leaving behind a hydrogenic

1s electron. Previous work on the FC method showed that for $0.910 < Z < Z_C^{\text{FC}}$ the outer electron still remains localised^[47,94]. It was proposed that this quasibound electron can tunnel through the barrier, formed by a combined effect of the long-range Coulomb repulsion and short range polarisation attraction in order to escape and produce a shape resonance^[47].

A more rigorous mathematical treatment of this effect is provided by Gridnev^[172], who shows that as the bound state approaches the threshold, it does not spread and will eventually become the bound state at the threshold value. Mathematically this means the Hamiltonian has an eigenstate even when $Z = Z_C^{\text{FC}}$ and there is no spread of the wavefunction.

A plot of $\langle r_1 \rangle$ and $\langle r_{12} \rangle$ is seen in Figure 5.1 calculated using the HF wavefunction, as a function of nuclear charge, with the HF critical nuclear charge Z_C^{HF} marked as the dotted line.

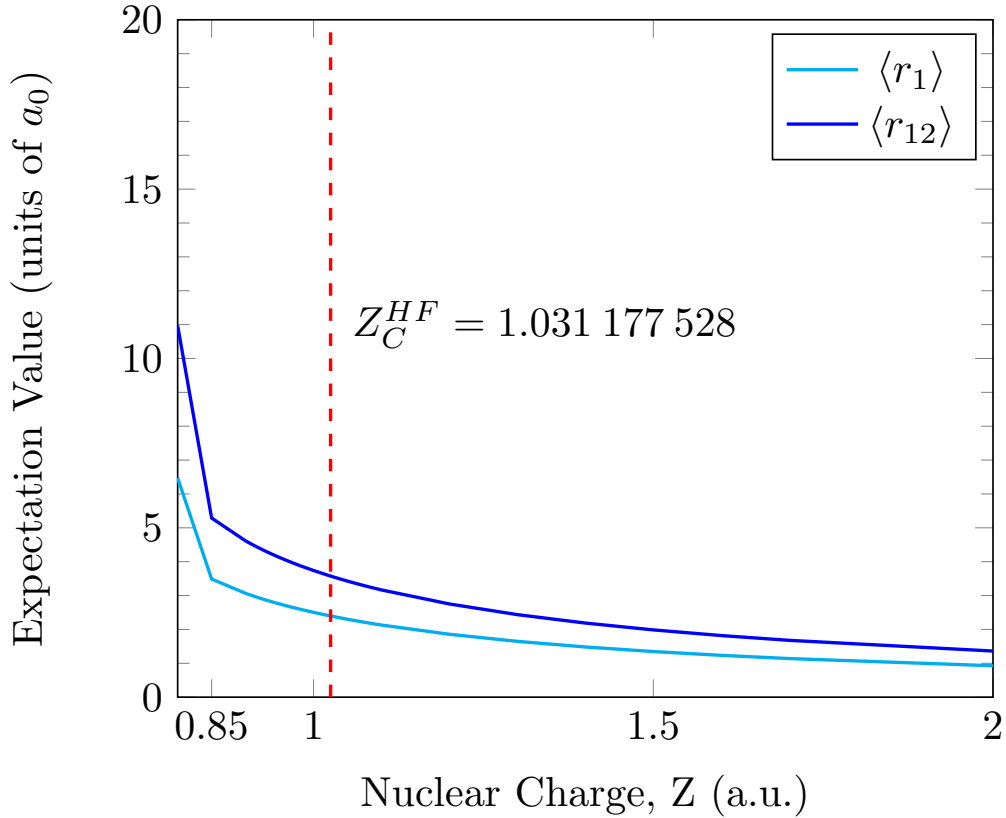


Figure 5.1 The expectation value of the nucleus-electron distance $\langle r_1 \rangle$, and the electron-electron distance $\langle r_{12} \rangle$, as a function of nuclear charge, Z . The HF critical nuclear charge value is shown by the vertical red dotted line.

Interestingly there is no behavioural change as Z is lowered just below the critical point which represents electron detachment, however below $Z = 0.85$ there is a sudden increase in $\langle r_1 \rangle$ and $\langle r_{12} \rangle$ which could be caused by a breakdown in accuracy

of the HF method. The HF implementation discussed in this work is valid for the restricted HF case, which might explain this result as it confines the electrons to be in the same orbital. The ill-converged HF expectation values for $Z < 0.85$ may reflect the inadequacy of the wavefunction to model one of the electrons as it becomes unbound.

The behaviour of $\langle r_1 \rangle$ and $\langle r_{12} \rangle$ below the critical charge point may also be explained by the physics underlying the situation and it could be highlighting a quasi-bound behaviour; indicating that the mean-field approach of HF allows the detaching electron to remain quasi-bound for a significant finite length scale. It could also be explained by a combined effect of these two possibilities, and warrants further investigation making for interesting future work.

Low nuclear charge energy convergence

In the previous section, it was shown that the HF results are invalid for $Z < 0.85$ due to poor convergence for the expectation values, $\langle r_1 \rangle$ and $\langle r_{12} \rangle$, but the convergence in the energy is now assessed at low nuclear charge. Table 5.10 shows HF energies calculated using 20 term wavefunctions for arbitrary charge systems where $Z < 1$, accurate to the number of digits presented. The convergence was assessed *via* looking for converged digits between a 20-term basis set and a 17 term basis set.

Table 5.10 The Hartree-Fock energy E_{HF} , for low nuclear charge systems, accurate to the number of digits presented. All energies are given in a.u.

Z	E_{HF}	Z	E_{HF}
0.9	-0.360 888 40	0.50	-0.053 4
0.85	-0.304 973 76	0.45	-0.035 1
0.80	-0.254 2	0.40	-0.020 7
0.75	-0.208 8	0.35	-0.010 2
0.70	-0.168 6	0.30	-0.004 8
0.65	-0.133 2	0.25	-0.004 2
0.60	-0.102 3	0.20	-0.003 7
0.55	-0.075 7	0.10	-0.002 4

The data in Table 5.10 shows that the energy convergence rapidly decreases from 8 significant figures for $Z = 0.85$ to just 4 significant figures when $Z = 0.8$; which confirms the charge limit for this HF method.

5.4 Electron density and Coulomb holes

Coulomb holes were calculated as the difference between the intracule densities obtained from the FC and HF methodologies. All intracule density calculations use a 4389 basis set size for the FC method and a 20 term basis set size for the HF method.

5.4.1 Helium atom and Lithium cation

In agreement with previous calculations, the Coulomb hole is negative for small r and positive for large r as a result of electron correlation increasing the probability of the electrons being apart at distances before and after the maxima in the intracule distributions. Figure 5.2 shows the FC and HF intracule densities along with the Coulomb holes for the lithium cation and helium. The inset graph reveals the secondary Coulomb hole for each system, with all key features (roots, area and minimum) tabulated in Table 5.11.

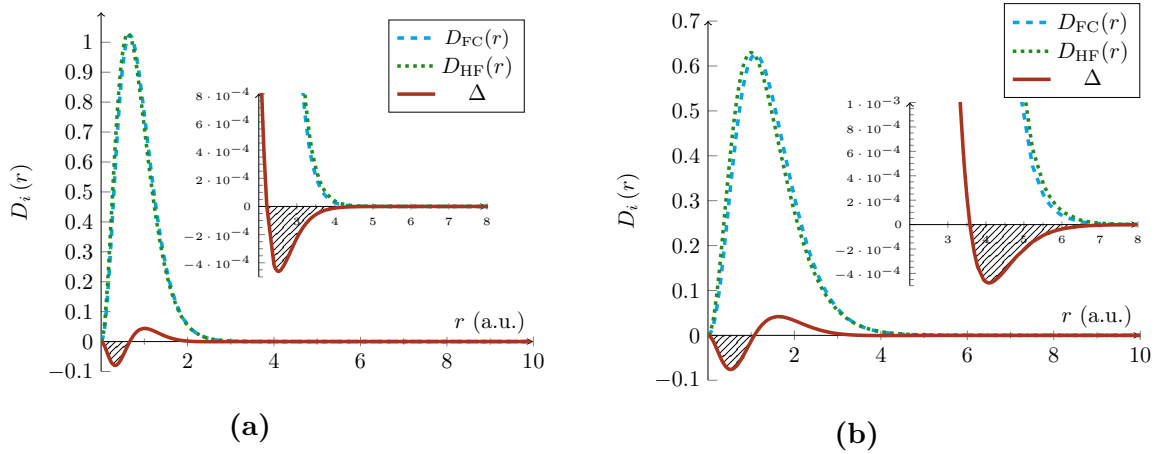


Figure 5.2 FC and HF intracule densities for (a) lithium cation and (b) helium atom. The solid line represents the difference between the two curves, the Coulomb hole. The inset plots shows the much smaller, secondary Coulomb hole for each system.

Table 5.11 Roots, area and minima of the Coulomb holes in the helium-like ions. The area of the Coulomb hole is measured in units of charge displaced, e .

System	Hole Number	Root 1	Root 2	Area	Minimum
Z_C	1	0	4.85	2.30×10^{-1}	1.91, -8.9×10^{-2}
H^-	1	0	3.20	1.33×10^{-1}	1.46, -7.46×10^{-2}
He	1	0	1.07	4.63×10^{-2}	0.526, -7.59×10^{-2}
	2	3.58	-	6.12×10^{-4}	4.08, -4.77×10^{-4}
Li^+	1	0	0.66	2.8×10^{-2}	0.325, -7.95×10^{-2}
	2	2.21	-	4.0×10^{-4}	2.516, -4.60×10^{-4}

In this work we confirm the existence of the secondary Coulomb hole, in agreement with the conclusions of Pearson *et al.*^[137]. The secondary Coulomb hole corresponds to a decrease in probability of finding the electrons far apart, which results from the intracule distribution having a second crossing point indicating that, although at shorter distances ($r_{12} < 3.6$ a.u. ≈ 1.9 Å for He and ≈ 1.17 Å for Li^+) the mean-field approach of HF allows electrons to get too close. At larger separations the electron correlation reduces the probability of two electrons being found at such distances.

Before exploring lower charge systems, the accuracies of the particle densities need to be assessed to check the validity of these results. The maximum error is calculated using a form appropriate to the method of calculation^[137], for the FC method

$$\left(\int_0^\infty [D_{4389}^{FC}(r) - D_{2856}^{FC}(r)]^2 dr \right)^{1/2}, \quad (5.56)$$

and for the HF method

$$\max_{r>0} |D_{20}^{HF}(r) - D_{15}^{HF}(r)|. \quad (5.57)$$

In equation (5.56), 2856 and 4389 represent the number of terms in the wavefunction and correspond to complete polynomials of order $\omega = 30$ and 35 , respectively, where $l+m+n \leq \omega$ which is discussed in Chapter 2. In equation (5.57), 15 and 20 represent the number of terms in the wavefunction for the Hartree-Fock calculations.

The maximum error in the FC helium intracule is calculated to be 2.616×10^{-9} and for the HF intracule 9.122×10^{-9} . In comparison, Gill *et al.* calculated their errors to be of the order 10^{-6} , so the methods employed in this work offer a significant increase in the accuracy of Coulomb hole calculations and their properties. A consequence

being that this confirms without a doubt the secondary Coulomb hole is not a numerical artefact.

5.4.2 Hydride anion and the critical nuclear charge system

Pearson *et al.*^[137] were unable to obtain satisfactory convergence in their hydride calculation but it has been previously calculated^[173]. With the improved accuracy of the methods employed in this work, it is possible to study Coulomb holes for this weakly bound system along with the critical nuclear charge system. Figure 5.3 shows the FC and HF intracule densities along with the Coulomb hole for the hydride anion and FC critical nuclear charge system Z_C^{FC} .

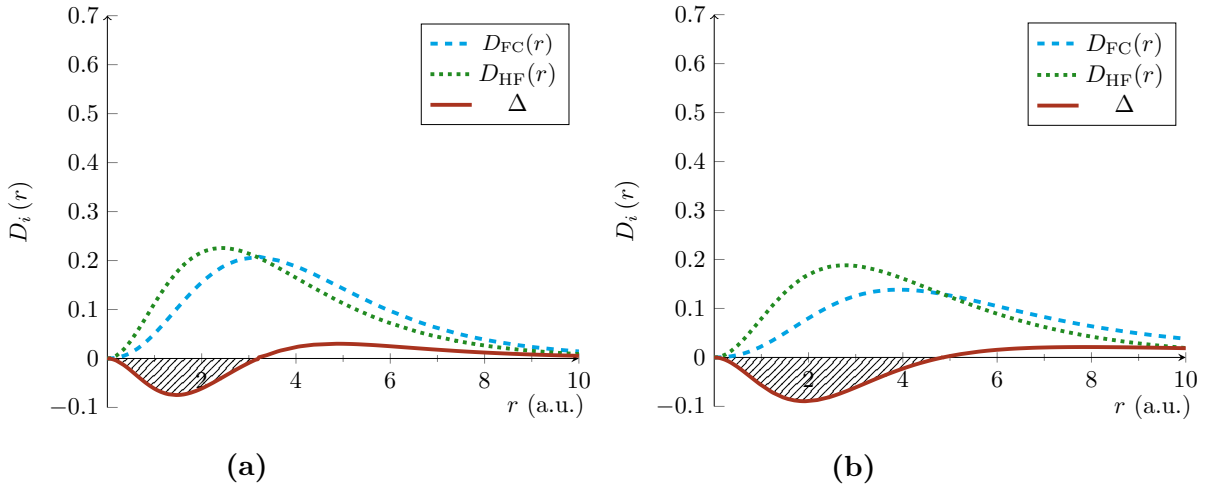


Figure 5.3 FC and HF intracule densities for the hydride anion and critical nuclear charge system Z_C^{FC} . The solid line represents the difference between the two curves, the Coulomb hole curve.

Figure 5.3 highlights how the effects of electron correlation are more pronounced for the anionic systems demonstrated by the much greater difference in the features of the FC and HF intracule densities when compared to those seen in Figure 5.2.

The intracules for the anionic systems are much more diffuse. The most probable separation (r_{max} , the maximum in the intracule) shifts from $2.43 a_0$ with HF to $3.17 a_0$ as a result of electron correlation, with the corresponding probability at these values showing a reduction by over 8 %. This is amplified further for Z_C^{FC} where r_{max} shifts from $2.77 a_0$ with the HF method to $3.92 a_0$ with the FC method, and the probability at r_{max} is decreased by over 25 % as a result of electron correlation in the correlated system.

By comparison, electron correlation in helium increases the most probable separation by about 10 % but the probability density at r_{max} changes by less than 1

%. The radius of the Coulomb hole for H^- is three times greater than for helium, and for Z_C^{FC} it is even greater. The Coulomb hole for H^- shows that the probability of the two electrons lying anywhere within a distance of $3.2 a_0$ from each other is less than it would be without correlation (i.e. under HF), and correspondingly the probability that the electrons are separated by more than $3.2 a_0$ is greater.

Each of the intracule densities are normalised to one which results in the net content of the hole equalling zero, but the total amount of charge displaced by the Coulomb hole is equal to the area of the curve between root 1 and root 2 seen in Table 5.11^[141]. For H^- , this is between $r_{12} = 0$ and $r_{12} = 3.2$ a.u. which corresponds to a charge displacement of $0.133e$; whereas for helium it is approximately one-twentieth of an electron at $0.0463e$.

5.4.3 Coulomb hole comparison

In order to compare the Coulomb holes calculated in this work, they have been scaled by the nuclear charge Z , plotted in Figure 5.4.

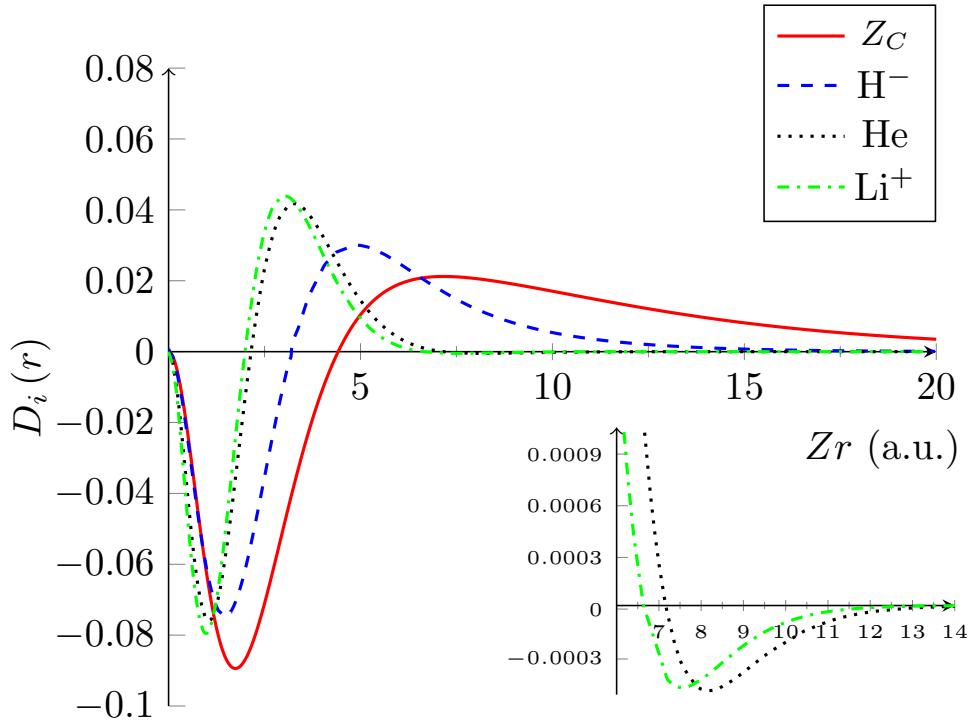


Figure 5.4 Z -scaled Coulomb holes for Z_C , H^- , He and Li^+

The anionic Coulomb holes exhibit a greater radius and displace a greater charge than helium or the cationic species which follows from the electron correlation effects being more pronounced in the anionic systems, and even at a distance of $10 a_0$ there is still a significantly high particle density. For helium and the lithium cation, the

Coulomb holes contract slightly toward the nucleus as the nuclear charge increases. This contraction is caused by the competition between the nucleus-electron attraction and the electron-electron repulsion, which for large Z systems, the nucleus-electron attraction dominates. This same effect is also seen for the secondary Coulomb holes for helium and the lithium cation seen in the inset of Figure 5.4.

5.4.4 Long range behaviour: Search for additional Coulomb holes

The short-range behaviour of the intracule densities reveals Coulomb holes, but equally interesting is the long-range behaviour of weakly bound systems with and without electron correlation. There is no evidence in the anionic H^- or Z_C systems, of a secondary Coulomb hole, and at all radial separations electron correlation increases the probability of the electrons being apart.

Even at $10 a_0$ the probability density is still large at 10^{-2} for H^- seen in Figure 5.5(b), contrary to helium where the probability density is of the order 10^{-9} at that radial separation, Figure 5.5(a). Correlation becomes increasingly important at larger separations, where beyond approximately $13.5 a_0$ the difference (solid line) becomes larger than the HF probability density (dotted line) Figure 5.5(b), and as the probability density in H^- approaches the value for the second Coulomb hole in helium, the Coulomb hole curve can be seen to align with the characteristics of the correlated intracule, Figure 5.6. This reveals that the statistically independent motion of the electrons in the HF approach increases the tendency of finding the electrons close together at short distances, whilst decreasing more rapidly the probability of them being far apart at large separations compared to when their motion is correlated.

As a secondary Coulomb hole has been proven to exist in this work for He and Li^+ , there is the potential for more holes at larger values of r , which is now investigated with confidence utilising the greater accuracy of the methods developed. For helium the errors in the intracule densities are of the order of a nano-hartree, which means results can only be trusted up to this point. Figure 5.5(a) shows a zoomed in view of the intracule densities down to a density scale of the order of calculated error.

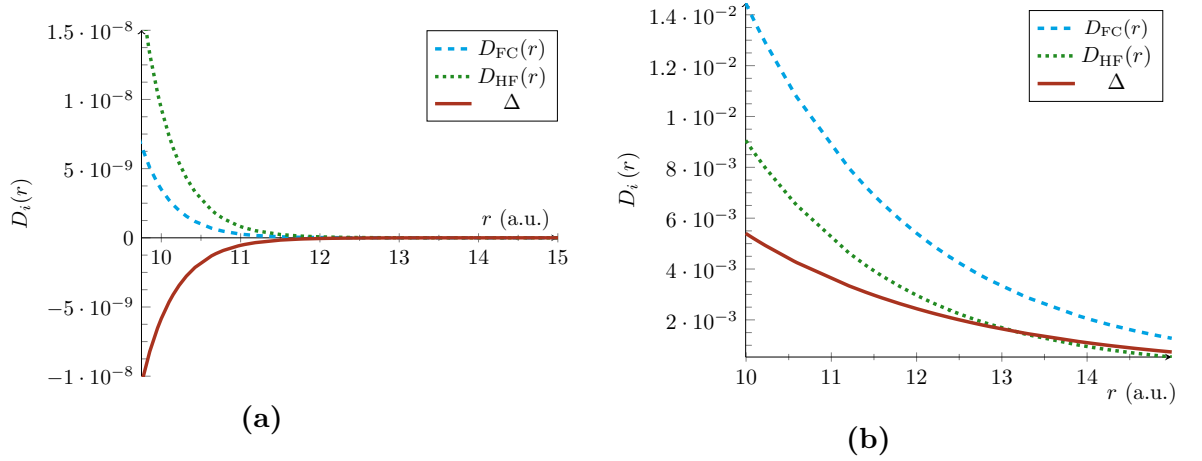


Figure 5.5 Long-range behaviour of the intracule distribution functions for (a) He and (b) H^- .

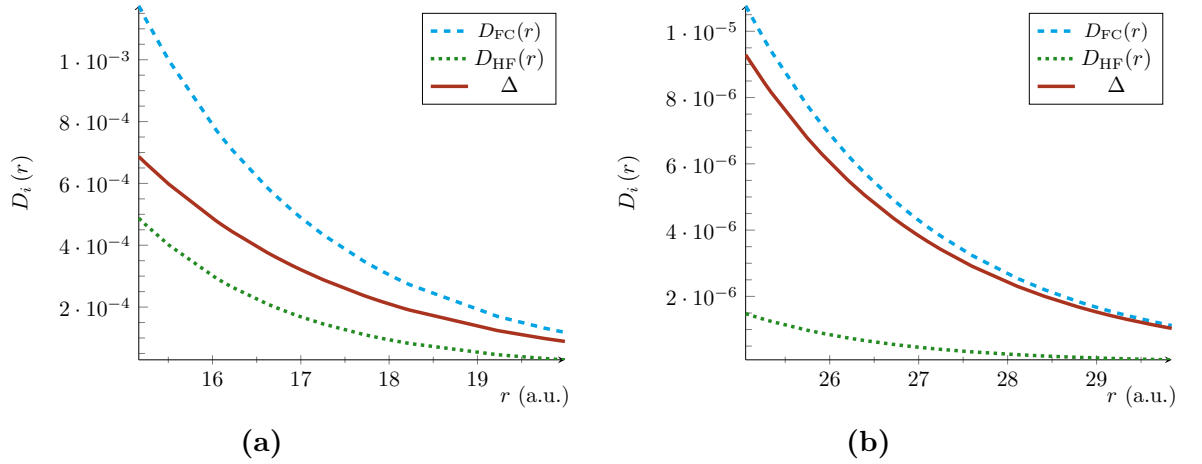


Figure 5.6 Long-range behaviour of the intracule distribution functions $D_i(r)$, $i = \text{FC}$ or HF , and the Coulomb hole function, Δ , for the hydride ion, H^- .

Within the accuracy used in this work, there is no evidence of additional Coulomb holes. The secondary Coulomb hole for helium is much smaller than the primary, so a tertiary hole will be even smaller and thus a greater accuracy may need to be employed in order to confidently disprove their existence.

5.5 Summary and Conclusions

A Hartree-Fock (HF) implementation has been discussed which uses a Laguerre-based wavefunction to investigate the ground state of two-electron atoms and ions within the clamped nucleus approximation. The one-electron integrals were processed using the efficient series solution method by exploiting the properties of Laguerre polynomials;

whilst the two-electron integrals are solved using an analytical form. A 20-term wavefunction is found to provide energies accurate to the pico-hartree or better for the helium isoelectronic sequence ($\text{He} \rightarrow \text{Ar}^{16+}$), whilst for the weaker bound hydride to one-tenth of a nano-hartree. The generated wavefunctions are proven to provide accurate inter-particle cusps, alongside virial conditions and expectation values. The energies calculated using this new HF implementation are compared with the energies calculated using the Fully Correlated (FC) method described in Chapter 2; resulting in accurate correlation energies.

The critical nuclear charge for which a two-electron system remains bound was investigated for the HF methodology by varying the nuclear charge, Z until the ionisation energy was zero, resulting in $Z_C^{HF}=1.031\,177\,528$. The radial behaviour near Z_C^{HF} was investigated using the expectation values of the inter-particle distances r_1 and r_{12} with results suggesting that the nucleus-electron and electron-electron distances vary smoothly as the nuclear charge decreases indicating that within the mean-field potential the electrons remain quasi-bound for $Z > 0.85$. Interesting future work would be to study this in greater detail with use of a stability analysis of the HF wavefunction to test for internal and external stability (i.e. with some constraints removed)^[174].

Coulomb holes were calculated as the difference between the intracule densities for the FC and HF methods. They were calculated for the lithium ion, helium atom, hydride anion and the system with the critical nuclear charge for binding two-electrons in the FC method in order to analyse the behaviour of the intracule density at long-range.

The errors in the intracule densities for the lithium ion and helium were calculated to be 3 orders of magnitude smaller than in previous work offering one of the most accurate analyses of Coulomb holes in the studied systems to date. It was found, as in previous work, that they exhibit a primary hole at small separations and a secondary hole at larger radial separations. Additional Coulomb holes were sought, but none were found within the accuracy of the FC and HF methods employed in this work.

The weaker bound hydride anion and critical nuclear charge systems showed only a primary Coulomb hole; with the inadequacies of HF exemplified by the large difference between FC and HF intracule densities. The long-range behaviour of a correlated system was not captured meaning the FC intracule density dominates the Coulomb hole behaviour at larger separations. Interesting future work could look into the emergence of the secondary Coulomb hole as the nuclear charge is increased

from the hydride anion which only exhibits a single Coulomb hole to that of helium which exhibits two.

All the results in this chapter and thesis are from solving the non-relativistic time-independent Schrödinger equation where it has been shown that as the nuclear charge increases, the effects of electron correlation gradually level off seen in Table 5.6^[175,176], however it has been reported that when including relativistic effects the repulsion of the electrons is overcome and the correlation energies continue to change^[177,178]. The inclusion of relativistic effects makes for challenging but very interesting future work.

CHAPTER 6



Excited S states of two electron atoms

6.1 Introduction

The work discussed in the previous chapters has centred around the ground state of three-particle atomic and molecular systems. Calculation of excited states present a more complicated problem however are important for interpreting results from spectroscopic measurements in astrochemistry and atmospheric chemistry, along with understanding of photochemical processes^[179]. Excited states also lie at the heart of modern and emerging technologies with a choice example being the recently developed strontium atomic clock^[180] which exploits the electronic excitation of supercold strontium atoms in optical traps to measure the accuracy of a second down to an uncertainty of 2×10^{-18} . Increased control of Rydberg states has garnered recent interest leading to possible new applications as logic gates in quantum computing to replace the conventional switch^[49].

A variety of techniques are used in conventional quantum chemistry to calculate excited states of atoms and molecules including post-Hartree-Fock methods such as Coupled Cluster (CC) theory^[68] and Configuration Interaction (CI.)^[68]. Another popular method to calculate excited states is with the use of time-dependent density functional theory (TDDFT) which offers one of the only tractable methods for calculating electronic excitation energies for systems which contain more than 20-30 atoms^[181]. The computationally intensive Complete Active Space Self Consistent Field (CASSCF) method is a multi-configurational method suited to calculating excited states of small-medium sized molecules^[182]. In this work, an excited state is defined as a system with principle quantum number, $n > 1$.

Pekeris' seminal work of 1958^[12] on the 1^1S ground state of helium is not accurate for highly excited singlet and triplet states, $n \gtrsim 4$, so he developed a modification with the addition of a second set of perimetric coordinates enabling accurate calculation of excitation energies^[50]. The radial distributions of low lying excited states of helium were investigated by Ugalde *et al.* where they showed that the inclusion of electron correlation led to a contraction of the Hartree-Fock electron clouds^[183]. In 1994 Drake

and Yan published very high accuracy variational energy calculations of the first 10 excited singlet and triplet states of helium under the clamped nucleus approximation using a double basis set of Hylleraas coordinates^[78]. In 1995, Burgers *et al.* used quantum defect theory to study the Rydberg series of helium in order to investigate the overlap of the inner and outer electrons in highly excited Rydberg states^[184]. Nakashima *et al.* presented the most accurate variational energy calculations for the excited singlet and triplet states of helium to date up to $n = 24$ with use of the free iterative-complement-interaction method which also includes nuclear motion^[185].

The original 3Body program can calculate excited S state eigenvalues as these are readily obtained from the diagonalisation of the secular determinant, where the first eigenvalue represents the ground state, E_0 , and subsequent eigenvalues the excited states, $E_n, n = 1, \dots, k$, where k is the number of terms in the wavefunction. As described in Section 2.2 it is necessary to enforce the constraint $\alpha = \beta$ to exploit the Laguerre orthogonality relation, meaning both particles are constrained by the same decay parameter which is not representative of the physical picture. Later in this chapter the limitation of this constraint is made clear by comparison to the case where $\alpha \neq \beta$.

Chapter summary

The aim of this chapter is to implement modifications to the series solution method to better describe excited states first proposed by Pekeris^[50], whilst extending the method by explicit inclusion of nuclear motion effects, 3 NLPs with no parameter constraints and much larger matrices. This is then shown to offer rapid convergence for excited state energies when compared to the original 3Body code. This new method and code will be used in future applications to study spectroscopic quantities. In this chapter the code will be used to provide a better understanding of the bound state stability of singlet and triplet states by studying the critical nuclear charge required for binding two electrons to a nucleus.

6.2 Method

The $\alpha = \beta$ constraint is not a hindrance for ground state calculations, ($n = 1$ for singlet and $n = 2$ for triplet), as the two fermions will occupy the same spatial region (singlet) or have a small spatial separation (triplet), thus imposing the condition is a valid representation of the physical situation and the long-range asymptotic behaviour. This is not true for excited states where each fermion can occupy different spatial regions, thus forcing α and β to be equal is not physically realistic.

6.2.1 Relaxing $\alpha = \beta$ NLP constraint

Consider the form of the wavefunction used in the ground state problem, reproduced here from equation (2.11)

$$\Psi(z_1, z_2, z_3) = e^{-\frac{1}{2}(\alpha z_1 + \beta z_2 + \gamma z_3)} \times \sum_{l,m,n=0}^{\infty} A(l, m, n) [L_l(\alpha z_1) L_m(\beta z_2) L_n(\gamma z_3) \pm L_l(\alpha z_2) L_m(\beta z_1) L_n(\gamma z_3)], \quad (6.1)$$

where $+$ corresponds to singlet states and $-$ to triplet states. The $\alpha z_1 + \beta z_2$ in the exponent only matches with the argument of the first two Laguerre terms in the first term in the bracketed expression (highlighted in blue), but do not match the second set (highlighted in red) invalidating the Laguerre orthogonality condition seen in equation (2.12). Orthogonality is restored when $\alpha = \beta$ is enforced (See chapters 2 and 3). To allow for the asymmetric behaviour between the fermions whilst exploiting the Laguerre orthogonality condition, it is necessary to seek a double exponent solution of the form^[50]

$$\Psi(r_1, r_2, r_{12}) = \left[e^{-\beta r_1 - \alpha r_2} G(r_1, r_2, r_{12}) \pm e^{-\beta r_2 - \alpha r_1} G(r_2, r_1, r_{12}) \right]. \quad (6.2)$$

Which in perimetric form is

$$\Psi(z_1, z_2, z_3) = \left[e^{-\frac{1}{2}(\alpha z_1 + \beta z_2 + \gamma z_3)} G(z_1, z_2, z_3) \pm e^{-\frac{1}{2}(\beta z_1 + \alpha z_2 + \gamma z_3)} G(z_2, z_1, z_3) \right]. \quad (6.3)$$

$G(r_1, r_2, r_{12})$ and $G(z_1, z_2, z_3)$ represents the basis in inter-particle and perimetric coordinates respectively. Pekeris introduced two sets of perimetric coordinates^[50], with the first being the standard form

$$\begin{aligned} u &= \alpha(r_2 + r_{12} - r_1) \\ v &= \beta(r_{12} + r_1 - r_2) \\ w &= \gamma(r_1 + r_2 - r_{12}), \end{aligned} \quad (6.4)$$

And the second set having the form

$$\begin{aligned} u' &= \alpha(r_1 + r_{12} - r_2) \\ v' &= \beta(r_{12} + r_2 - r_1) \\ w' &= \gamma(r_1 + r_2 - r_{12}). \end{aligned} \quad (6.5)$$

The permutation of r_1 and r_2 is accomplished by the interchange of $u \rightleftharpoons u'$ and $v \rightleftharpoons v'$ enforcing the required fermionic exchange. As a basis for the expansion of Ψ Pekeris used^[50]

$$f(l, m, n) = e^{-\frac{1}{2}(u+v+w)} L_l(u) L_m(v) L_n(w) \pm e^{-\frac{1}{2}(u'+v'+w')} L_l(u') L_m(v') L_n(w'). \quad (6.6)$$

The full wavefunction takes the form

$$\Psi = \sum_{l,m,n=0}^{\infty} C(l, m, n) f(l, m, n), \quad (6.7)$$

where $C(l, m, n)$ represent the coefficients. To keep consistent with the z_i perimetric form used throughout this thesis, u, v, w, u', v', w' are expressed in $z_1, z_2, z_3, z'_1, z'_2, z'_3$ where

$$\begin{aligned} u &= \alpha z_1, & u' &= \alpha z_2 = \alpha z'_1 \\ v &= \beta z_2, & v' &= \beta z_1 = \beta z'_2 \\ w &= \gamma z_3, & w' &= \gamma z_3 = \gamma z'_3 \end{aligned} \quad (6.8)$$

Resulting in

$$\begin{aligned} \Psi(z_1, z_2, z_3) &= \sum_{l,m,n=0}^{\infty} C(l, m, n) \\ &\times \left[e^{-\frac{1}{2}(\alpha z_1 + \beta z_2 + \gamma z_3)} L_l(\alpha z_1) L_m(\beta z_2) L_n(\gamma z_3) \right. \\ &\left. \pm e^{-\frac{1}{2}(\alpha z'_1 + \beta z'_2 + \gamma z'_3)} L_l(\alpha z'_1) L_m(\beta z'_2) L_n(\gamma z'_3) \right]. \end{aligned} \quad (6.9)$$

Unlike in the single exponent form of equation (6.1), the arguments of each set of Laguerre functions has a matching argument in its own exponential factor (highlighted in blue and red) allowing for α and β to be unique whilst satisfying the Laguerre orthogonality relation. A consequence of this is that it is effectively the sum of two wavefunctions, so the symmetric and antisymmetric numbering schemes cannot be exploited when building the matrices. As in the ground state method, the spin state is controlled by the \pm with $+$ representing singlet states and $-$ triplet states.

6.2.2 Series solution

Analogous to the ground state methodology the properties of the Laguerre polynomials can be used to express the matrix coefficients in terms of the values of

the recursion relation coefficients $C(l, m, n)$. The Jacobian for the transformation between the Cartesian coordinates and inter-particle coordinates is $r_1 r_2 r_{12}$

$$S = r_1 r_2 r_{12} (H\Psi - E\Psi) = \sum_{l,m,n=0}^{\infty} R(l, m, n) f(l, m, n), \quad (6.10)$$

where

$$R(l, m, n) = \sum_{a,b,c=-2}^{+2} C(l+a, m+b, n+c) R_{a,b,c}(l, m, n), \quad (6.11)$$

with $R(l, m, n)$ expressed in terms of the $C(l, m, n)$ coefficients in the wavefunction, equation (6.7). The orthogonality of $f(l, m, n)$ to S yields¹

$$\sum_{l,m,n=0}^{\infty} R(l, m, n) T(l, m, n, l', m', n') = S(l', m', n') = 0. \quad (6.12)$$

The function T has the form^[50]

$$T(l, m, n, l', m', n') = \delta(n, n') \left\{ \delta(l, l') \delta(m, m') + \frac{(-)^{m+m'} 4k}{(1+k)^2} \times \left(\frac{k-1}{k+1} \right)^{l+l'+m+m'} g(l, m') g(l', m) \right\}, \quad (6.13)$$

where $k = \beta/\alpha$, $x = [(k+1)/(k-1)]^2$ and $g(l, m)$ represent Gauss hypergeometric functions^[187]

$$g(l, m) = \sum_{r=0}^{\min(l,m)} \frac{(-x)^r (l+m-r)!}{r! (l-r)! (m-r)!}, \quad (6.14)$$

and the $S(l', m', n')$ are homogenous linear equations in the $C(l, m, n)$, the vanishing of whose determinant yields the energy eigenvalue ϵ .

As in the ground state method each matrix element consists of a sum of coefficients, C_{ik} , where i and k denote the row and column of the matrix. Substituting $i(\equiv l, m, n)$ and $k(\equiv l+a, m+b, n+c)$ into the recursion relation results in

$$\sum_k R_{ik} B_k = 0, \quad (6.15)$$

where B_k is the eigenvector, and the problem is then solved as a generalised eigenvalue problem.

¹This orthogonalisation procedure can be confirmed by explicitly conducting the integration of $\Psi(H-E)\Psi r_1 r_2 r_{12} = 0$ using the correct volume element^[186].

6.2.3 Implementation

Chapter 3 describes the computational methods used, whereas in this section an overview of how the theory was implemented is presented. It is administered as a modification to the ground state method detailed in Chapter 2, where the determinant is written as

$$r_1 r_2 r_{12} (\hat{H}\psi - E\psi) = \sum_{a,b,c=-2}^2 A(l+a, m+b, n+c) C_{a,b,c}(l, m, n) = 0. \quad (6.16)$$

This represents the solution to the three-particle Schrödinger equation for the ground state problem. The sparsity of the matrices is achieved by applying the [three](#) Laguerre orthogonality conditions represented as Kronecker delta functions in equation (6.17), to each of the terms in the recursion relation resulting in a large number of matrix elements being set to zero. The matrix coefficients can be expressed as

$$\sum_{a,b,c=-2}^2 C_{a,b,c}(l-a, m-b, n-c) \delta(n, n') \delta(m, m') \delta(l, l'). \quad (6.17)$$

The condition of orthogonality in the excited state problem is different than the ground state, resulting in a new expression for the matrix element coefficients

$$\sum_{a,b,c=-2}^2 C_{a,b,c}(l-a, m-b, n-c) \delta(n, n') [\delta(m, m') \delta(l, l') \pm T(l, m, n, l', m', n')]. \quad (6.18)$$

To aid in implementation, the function T in equation (6.13) is expanded

$$T(l, m, n, l', m', n') = \delta(n, n') \delta(m, m') \delta(l, l') + \frac{(-)^{m+m'} 4k}{(1+k)^2} \left(\frac{k-1}{k+1} \right)^{l+l'+m+m'} \times g(l, m') g(l', m) \delta(n, n') \quad (6.19)$$

To make the motivation for this expansion clearer, the following substitution is made in equation (6.19) to isolate the Kronecker delta terms

$$\xi = \frac{(-)^{m+m'} 4k}{(1+k)^2} \left(\frac{k-1}{k+1} \right)^{l+l'+m+m'} g(l, m') g(l', m). \quad (6.20)$$

Equation (6.19) becomes

$$T(l, m, n, l', m', n') = \underbrace{\delta(n, n') (m, m') \delta(l, l')}_{\text{Original}} + \underbrace{\xi \delta(n, n')}_{\text{New}}. \quad (6.21)$$

When the “Original” triple product of Kronecker delta functions is multiplied by the recursion relation, it will result in the matrices used to solve the ground state problem which we already have. The second “New” quantity represents the $\{n, n'\}$ orthogonality relation multiplied by ξ

$$\xi \int_0^\infty e^{-x} L_n(x) L_{n'}(x) dx = \xi \delta_{n,n'} \begin{cases} \xi, & n = n' \\ 0, & n \neq n' \end{cases} \quad (6.22)$$

To apply this single orthogonality condition, the original 57-term recursion relation is filtered as the $\{l, l'\}$ and $\{m, m'\}$ orthogonality conditions are not required for the “New” factor. This results in “New” consisting of 5 terms representing unique $\{n + c\}$ terms

$$\{n - 2, n - 1, n, n + 1, n + 2\}, \quad (6.23)$$

whilst the “Original” term still considers all unique combinations of $\{l + a, m + b, n + c\}$ as before; meaning the total number of terms in the recursion relation is still 57 as no new terms have been added.

This explains the loss of sparsity in the matrices **H** and **S** seen in Figure 3.4 of Chapter 3. The “Original” term forms sparse matrix representations, but the addition of the “New” terms formed by $\xi \delta(n, n')$ “switches back on” matrix elements set to zero by the other two orthogonality conditions, $\{l, l'\}$ and $\{m, m'\}$, resulting in a dense secular determinant. Another unfortunate consequence is that each of the 5 coefficients from $\xi \delta(n, n')$ are very large and cause the size of individual algebraic non-zero matrix elements to increase dramatically which increases the required RAM and storage size.

6.2.3.1 Inclusion of nuclear motion

In this work the theory outlined above has been extended to include the effects of nuclear motion by using a finite nuclear mass, M in the Hamiltonian. As this chapter is dedicated to the treatment of two electron atoms, the Hamiltonian for a two-electron atom of the form $\{e^- e^- M^{Z+}\}$ is then

$$\hat{H} = -\frac{1}{2} \nabla_1^2 - \frac{1}{2} \nabla_2^2 - \frac{1}{2M} \nabla_3^2 - \frac{Z}{r_1} - \frac{Z}{r_2} + \frac{1}{r_{12}}. \quad (6.24)$$

Where M is the mass of the nucleus and Z the nuclear charge with r_1, r_2 the distances of the first and second electrons from the nucleus. This Hamiltonian form is used when generating the 57-term recursion relation using the techniques outlined in Chapter 2 and 3. Chapter 3 discusses the computational cost of leaving all variables

undefined when the matrices are generated, and in order to shrink their byte size the electron masses and charges are set to 1 and -1 respectively when the matrices are generated, i.e. ($m_1 = m_2 = 1, Z_1 = Z_2 = -1$). Only M, Z and the three NLPs, α, β, γ are left undefined in the generated matrices meaning they are valid for atomic systems with infinite or finite nuclear mass and charge values.

6.2.3.2 Inclusion of three NLPs, α, β, γ

The correct long-range, asymptotic behaviour for atoms in inter-particle coordinates is given by^[7,53]

$$e^{-A(r_1+r_2)}. \quad (6.25)$$

In his original work Pekeris enforced this via the exponential form^[12]

$$e^{-\sqrt{E}(r_1+r_2)}. \quad (6.26)$$

For excited states the forms in equations (6.25) and (6.26) are not satisfactory as both r_1 and r_2 are being multiplied by the same decay parameter A , instead

$$e^{-(Ar_1+Br_2)}, \quad (6.27)$$

is desired. To achieve this form Pekeris used the following substitutions for α, β and γ in the wavefunction^[50]

$$\alpha = (2\epsilon^2 - Z^2)^{1/2}, \quad \beta = Z, \quad \gamma = \alpha + \beta. \quad (6.28)$$

Where ϵ^2 is the energy eigenvalue. In the current work, these do not need to be assumed as all three parameters, α, β, γ are left undefined in the recursion relation and matrices, so can be optimised in order to achieve more rapid energy convergence; with the exponential factor taking the form

$$e^{-Ar_1-Br_2-Cr_{12}} \implies \alpha \neq \beta, \gamma \neq \alpha + \beta. \quad (6.29)$$

Allowing α, β and γ to be unique offers a great advantage over the ground state methodology as each electron can now act independently allowing for excited states and diffuse systems to be studied with greater accuracy.

6.2.4 Variational principle for excited states

The variational theorem is routinely applied to ground state problems but is readily extendible to excited states albeit with added constraints. Equation (2.34) states the variational theorem with respect to the ground state energy, E_0 , and by the Hylleraas, Undheim, Macdonald (HUM) theorem^[188,189] the remaining eigenvalues are also upper bounds to the exact excited state energies, E_1, E_2, \dots, E_n if the spectrum is bounded from below; and the approximate excited state wavefunction is orthogonal and non-interacting with all approximate wavefunctions of lower excited states and the ground state. These conditions prevent variational collapse for excited states with the same symmetry as the ground state. The HUM theorem is satisfied automatically in this work and by several quantum chemical models including configuration interaction (CI) techniques that use a single configurational reference state (e.g. Hartree-Fock)^[190].

Löwdin showed that in order to obtain an accurate excited state wavefunction, explicit orthogonalisation of excited state wavefunctions with the ground state wavefunction is not strictly necessary; by analysing the error expression for the first excited state and showing it can be minimised by making the second root of the secular determinant, J_1 , sufficiently close to E_1 without requiring $J_0 = E_0$ ^[191]. Related to this point, Drake *et al.* calculate each excited state for helium using different basis functions^[78] meaning the individual states may not necessarily be orthogonal to one another^[185]. Pekeris showed that his method for excited states is variational^[50] thus satisfying the HUM theorem, and the agreement is tested for the modified theory in this chapter using the following orthogonality condition

$$\langle \Psi_i | \Psi_j \rangle = \int_0^\infty \Psi_i^* \Psi_j dV, \quad i \neq j, \quad (6.30)$$

to assess agreement with zero. Two separate calculations are conducted:

1. Where Ψ_i and Ψ_j are obtained from a single diagonalisation.
2. Where Ψ_i and Ψ_j are optimised individually.

First the orthogonality condition for excited states calculated using a single diagonalisation, and secondly one where each state is optimised individually.

6.3 Results and Discussion

6.3.1 Energy convergence

Energy convergence for the excited state method is assessed by increasing the matrix order ω and determining converged digits, i.e. digits that do not change as ω is increased; and by comparing with literature. Extensive convergence data has been collected for a large number of excited states of helium with the nucleus clamped and in motion for every ω value up to $\omega = 25$. This corresponds to a 3276×3276 size matrix and represents the maximum order of the matrix used for the energy calculations in this chapter. More accurate results would be expected to be obtained using a larger matrix size, but requires a greater amount of time and computational resources. The numerous calculated energy values cannot all be listed so a selection of singlet and triplet helium excited states have been chosen, n^1S , n^3S for $n = 2, 4, 6, 8, 10, 12$; with energy values calculated using ω values of 5, 10, 15, 20, 25 for clamped and finite nuclear mass cases. For highly excited states, $n > 6$ small matrix sizes calculate the energy to be greater than the threshold energy, $E > E_{th}$

$$E_{th} = -\frac{Z^2}{2} \frac{1}{1 + M^{-1}}, \quad (6.31)$$

and so have been omitted due to the basis size being inadequate to calculate the system as being bound. The paper by Nakashima *et al.* is the reference used for comparison with the energies calculated in this work, in which they used the CODATA 2006^[31] value for the nuclear mass of helium, $m_{He} = 7294.299\,536\,5$ a.u. whereas in this work the CODATA 2014^[52] nuclear mass value is used, $m_{He} = 7294.299\,541\,36$ a.u.

Table 6.1 Energy convergence data for the excited 2^1S , 4^1S , 6^1S , 8^1S , 10^1S , 12^1S states of helium using three NLPs, α , β , γ with clamped nucleus and nucleus in motion. Dashes (-) represent systems where $E > E_{\text{th}}$.

2^1S				4^1S			
ω	Matrix Size	$M = \infty$	Energy/ E_h	$M = m_{\text{He}}$	$M = \infty$	Energy/ E_h	$M = m_{\text{He}}$
5	56	-2.145 956 413 475 48	-2.145 660 948 066 83	-2.033 523 185 054 90	-2.033 523 185 054 90	-2.033 244 689 853 48	-2.033 244 689 853 48
10	286	-2.145 973 968 843 89	-2.145 678 539 117 82	-2.033 585 387 693 85	-2.033 585 387 693 85	-2.033 306 675 642 28	-2.033 306 675 642 28
15	816	-2.145 974 044 392 57	-2.145 678 586 181 23	-2.033 586 669 296 45	-2.033 586 669 296 45	-2.033 307 768 074 31	-2.033 307 768 074 31
20	1771	-2.145 974 046 014 43	-2.145 678 587 580 03	-2.033 586 712 166 38	-2.033 586 712 166 38	-2.033 307 817 452 94	-2.033 307 817 452 94
25	3276	-2.145 974 046 051 59	-2.145 678 587 580 34	-2.033 586 717 001 84	-2.033 586 717 001 84	-2.033 307 817 480 71	-2.033 307 817 480 71
Lit. Energy ^[185]		-2.145 974 046 054 4	-2.145 678 587 580 57	-2.033 586 717 030 73	-2.033 586 717 030 73	-2.033 307 817 481 29	-2.033 307 817 481 29
6^1S				8^1S			
ω	Matrix Size	$M = \infty$	Energy/ E_h	$M = m_{\text{He}}$	$M = \infty$	Energy/ E_h	$M = m_{\text{He}}$
5	56	-2.014 490 851 626 02	-2.014 214 728 641 70	-	-	-	-
10	286	-2.014 557 112 596 21	-2.014 277 607 064 46	-2.008 080 227 165 61	-2.008 080 227 165 61	-2.007 805 987 297 67	-2.007 805 987 297 67
15	816	-2.014 562 629 183 08	-2.014 286 464 783 85	-2.008 092 112 966 55	-2.008 092 112 966 55	-2.007 816 836 029 20	-2.007 816 836 029 20
20	1771	-2.014 563 071 105 47	-2.014 286 898 256 02	-2.008 093 503 399 86	-2.008 093 503 399 86	-2.007 818 227 445 25	-2.007 818 227 445 25
25	3276	-2.014 563 098 100 42	-2.014 286 911 006 82	-2.008 093 620 634 56	-2.008 093 620 634 56	-2.007 818 345 911 85	-2.007 818 345 911 85
Lit. Energy ^[185]		-2.014 563 098 446 62	-2.014 286 911 184 42	-2.008 093 622 105 61	-2.008 093 622 105 61	-2.007 818 346 203 25	-2.007 818 346 203 25
10^1S				12^1S			
ω	Matrix Size	$M = \infty$	Energy/ E_h	$M = m_{\text{He}}$	$M = \infty$	Energy/ E_h	$M = m_{\text{He}}$
5	56	-	-	-	-	-	-
10	286	-2.005 124 659 196 96	-2.004 849 794 968 22	-	-	-	-
15	816	-2.005 140 463 257 92	-2.004 865 598 569 86	-2.003 551 633 017 81	-2.003 551 633 017 81	-2.003 277 401 582 95	-2.003 277 401 582 95
20	1771	-2.005 142 612 910 89	-2.004 867 749 815 21	-2.003 553 855 357 71	-2.003 553 855 357 71	-2.003 279 293 501 99	-2.003 279 293 501 99
25	3276	-2.005 142 940 887 63	-2.004 868 103 995 71	-2.003 554 614 321 13	-2.003 554 614 321 13	-2.003 279 961 420 75	-2.003 279 961 420 75
Lit. Energy ^[185]		-2.005 142 991 747 99	-2.004 868 128 875 83	-2.003 554 625 012 23	-2.003 554 625 012 23	-2.003 279 983 605 862 31	-2.003 279 983 605 862 31

Table 6.2 Energy convergence data for the excited 2^3S , 4^3S , 6^3S , 8^3S , 10^3S and 12^3S states of helium using three NLPs, α, β, γ with clamped nucleus and nucleus in motion. Dashes (-) represent systems where $E > E_{\text{th}}$.

2^3S				4^3S			
ω	Matrix Size	$M = \infty$	Energy/ E_h	$M = m_{\text{He}}$	Energy/ E_h	$M = \infty$	$M = m_{\text{He}}$
5	56	-2.175 229 224 254 17	-2.174 930 045 708 933 2	-2.036 511 359 191 96	-2.036 232 104 304 42	-2.036 511 359 191 96	-2.036 232 104 304 42
10	286	-2.175 229 378 149 86	-2.174 930 190 625 503 8	-2.036 512 072 104 56	-2.036 232 816 785 93	-2.036 512 072 104 56	-2.036 232 816 785 93
15	816	-2.175 229 378 235 62	-2.174 930 190 711 391 4	-2.036 512 082 837 81	-2.036 232 827 528 35	-2.036 512 082 837 81	-2.036 232 827 528 35
20	1771	-2.175 229 378 236 790	-2.174 930 190 712 307 3	-2.036 512 083 090 34	-2.036 232 827 763 02	-2.036 512 083 090 34	-2.036 232 827 763 02
25	3276	-2.175 229 378 236 791 1	-2.174 930 190 712 309 3	-2.036 512 083 098 12	-2.036 232 827 788 55	-2.036 512 083 098 12	-2.036 232 827 788 55
Lit. Energy ^[185]		-2.175 229 378 236 791 3	-2.174 930 190 712 309 5	-2.036 512 083 098 24	-2.036 232 827 788 69	-2.036 512 083 098 24	-2.036 232 827 788 69
6^3S				8^3S			
ω	Matrix Size	$M = \infty$	Energy/ E_h	$M = m_{\text{He}}$	Energy/ E_h	$M = \infty$	$M = m_{\text{He}}$
5	56	-2.015 346 549 715 68	-2.015 070 243 040 36	-2.008 425 013 134 01	-2.008 149 686 714 76	-	-
10	286	-2.015 377 394 317 25	-2.015 101 107 328 56	-2.008 427 114 822 72	-2.008 151 799 290 83	-2.008 427 114 822 72	-2.008 151 799 290 83
15	816	-2.015 377 447 997 77	-2.015 101 162 954 94	-2.008 427 120 876 93	-2.008 151 805 355 34	-2.008 427 120 876 93	-2.008 151 805 355 34
20	1771	-2.015 377 452 961 19	-2.015 101 168 083 06	-2.008 427 121 998 60	-2.008 151 806 326 36	-2.008 427 121 998 60	-2.008 151 806 326 36
25	3276	-2.015 377 452 991 04	-2.015 101 168 230 51	-2.008 427 122 064 72	-2.008 151 806 543 95	-2.008 427 122 064 72	-2.008 151 806 543 95
Lit. Energy ^[185]		-2.015 377 452 992 86	-2.015 101 168 236 72	-2.008 427 122 064 72	-2.008 151 806 543 95	-2.008 427 122 064 72	-2.008 151 806 543 95
10^3S				12^3S			
ω	Matrix Size	$M = \infty$	Energy/ E_h	$M = m_{\text{He}}$	Energy/ E_h	$M = \infty$	$M = m_{\text{He}}$
5	56	-	-	-	-	-	-
10	286	-2.005 294 464 332 76	-2.005 019 558 699 37	-	-	-	-
15	816	-2.005 310 757 558 21	-2.005 035 874 881 59	-2.003 650 053 228 4	-2.003 373 331 402 9	-2.003 650 053 228 4	-2.003 373 331 402 9
20	1771	-2.005 310 791 889 56	-2.005 035 909 168 44	-2.003 650 208 672 6	-2.003 375 561 936 0	-2.003 650 208 672 6	-2.003 375 561 936 0
25	3276	-2.005 310 794 891 40	-2.005 035 911 284 80	-2.003 650 606 331 42	-2.003 375 961 403 44	-2.003 650 606 331 42	-2.003 375 961 403 44
Lit. Energy ^[185]		-2.005 310 794 915 61	-2.005 035 912 200 71	-2.003 650 626 784 73	-2.003 375 974 060 73	-2.003 650 626 784 73	-2.003 375 974 060 73

The accuracy of the excited state energies is excellent for both clamped nucleus and nucleus in motion, especially for low lying excited states. Singlet $2^1S, 4^1S$ are converged to 13 and 12 significant figures respectively for nucleus in motion and triplet $2^3S, 4^3S$ are converged to 16 and 13 significant figures respectively for nucleus in motion. The current implementation of the excited state Python code is only able to run at machine double precision (16 digits, binary 64 although parts of NumPy are able to use C long-doubles) which means the precision of the ground triplet state, 2^3S , has reached the maximum that is possible without extending the program to quadruple or arbitrary precision, making for important future work. When compiling LAPACK for use with Python and the excited state code it is crucial not to compile it against the Mac Accelerate framework bindings due to excessive numerical instability issues occurring with various matrix operations.

The 2^1S state with nucleus in motion is converged to 13 significant figures whereas the 12^1S state is converged to 8 significant figures. This decrease in accuracy is hypothesised to originate from more Laguerre functions being required to “fit” to the true wavefunctions of higher excited states due to their diffuse nature. Future work would see larger matrices generated ($\omega \geq 30$) which should offer excellent convergence across a wide range of excited states of two-electron atoms.

When using a small number of basis functions, e.g. 56 or 286 for highly excited states such as $n^1S, n^3S, n = 8, \dots, 12$ seen in Tables 6.1 and 6.2 the optimisation is difficult to complete, as the energy is calculated to be above the threshold energy of the system, equation (6.31). The optimisation will continually decrease α or β in order to minimise the energy and eventually the calculation crashes as the electron is effectively detached from the system. The ground state method can incorrectly optimise unbound systems as by enforcing $\alpha = \beta$ averages out the effect of the detaching electron with the inner electron in the optimisation calculation.

6.3.2 Effect of principle quantum number, n , on NLPs, α, β, γ

To enforce the correct asymptotic, long-range behaviour Pekeris used the expressions in equation (6.28) for α, β and γ . In this work all three are optimised individually, and Figure 6.1 compares the optimised values calculated in this work against Pekeris’ values for the n^1S states, $n = 2, \dots, 15$ of helium using a 3276 size matrix with clamped nucleus.

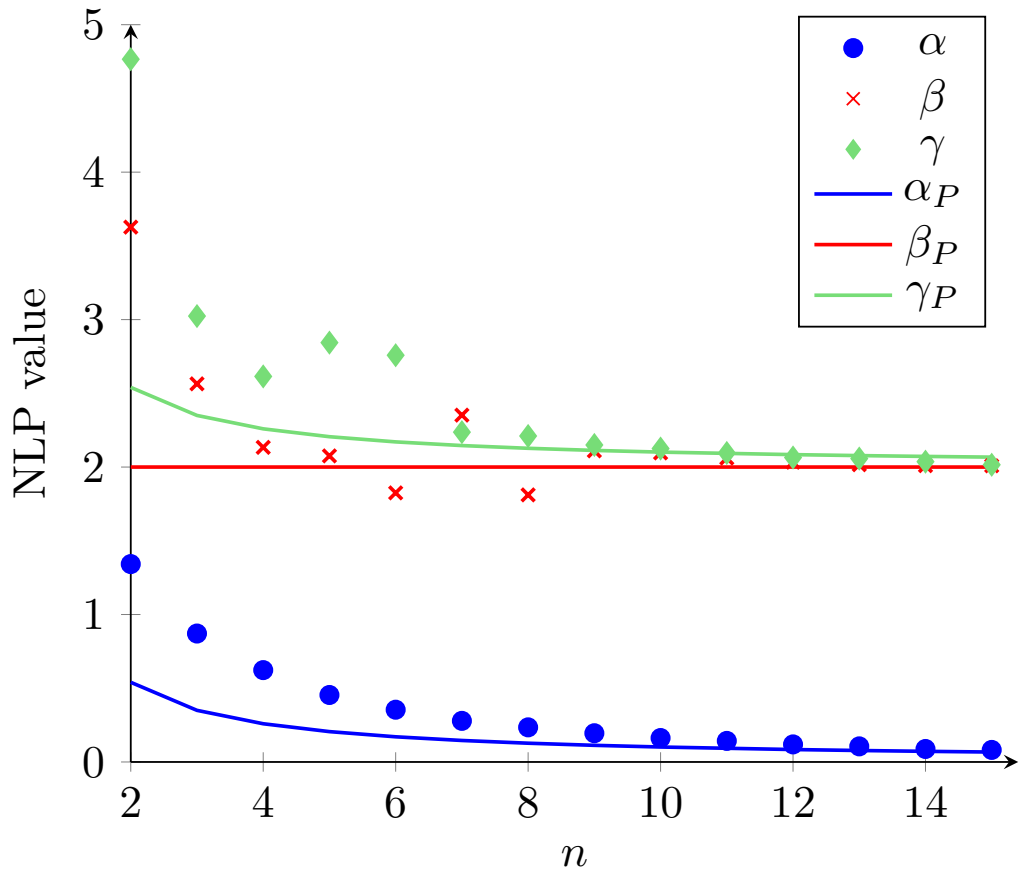


Figure 6.1 Effect of principle quantum number, n on the NLP values for the n^1S states of helium, $n = 2, \dots, 15$ using a 3276 size matrix and clamped nucleus. The solid lines α_P , β_P and γ_P represent the asymptotic forms used by Pekeris given in equation (6.28).

As the principle quantum number, n , increases, Figure 6.1 shows that the optimised NLP values and Pekeris' values become more similar. In the high n limit the expressions used by Pekeris offer a good representation of the system; as the excited electron at a high n will occupy a very diffuse spatial region allowing the hydrogenic-like behaviour of the nucleus and inner electron to increase, which is described by setting $\beta = Z$, i.e. hydrogenic asymptotic form, e^{-Zr} .

The optimisation of α and γ experiences fluctuations for mid-range principle quantum numbers, which could be caused by a number of reasons including the applicability of the Nelder-Mead algorithm used for the optimisation. Another cause could be due to the very flat nature of the potential energy surface making optimisation difficult using three NLPs as multiple combinations of α , β and γ achieve similar results. It should be noted that there is a single minimum on the PE surface for the excited atomic states for either case of $\alpha < \beta$ or $\beta > \alpha$, as was found with the ground state where, $\alpha = \beta$. Future work will require a thorough investigation into

this behaviour by trialling 1, 2 and 3 NLPs along with other optimisation techniques to develop a protocol for these optimisation calculations. All the energies calculated using the optimised NLPs showed better agreement with literature values compared to those using the expressions in equation (6.28). The behaviour displayed by the optimisation of the β parameter compared to α suggests that there are multiple values of β that will produce the same energy result whilst α shows a strong consistent convergence trend. It can be hypothesised that fixing β in the calculation as in equation (6.28), will not cause a great loss in energy convergence, whilst reducing the computational complexity.

6.3.3 Effect of nuclear motion

To assess the effect of nuclear motion on the excited states, the percentage difference is calculated between the energies of the fixed nucleus calculation, E_∞ , and the nucleus in motion calculation, E

$$E_{nm}(\%) = \left(\frac{E_\infty - E}{E} \right) \times 100, \quad (6.32)$$

where, $E_{nm}(\%)$ is the effect of nuclear motion, shown in Tables 6.3 and 6.4 for a range of excited states. This is more representative than calculating the absolute difference, $\Delta E = E_\infty - E$, as the magnitude of the change is relative to the system energy.

Table 6.3 Energies calculated for the excited singlet states of helium with clamped nucleus, E_∞ and nucleus in motion, E . The helium mass is taken from CODATA 2014^[52], $M = 7294.299\,541\,36$ a.u. Digits listed are converged.

State ^a	E	E_∞	$E_{nm}(\%)$
1^1S	-2.903 304 557 729	-2.903 724 377 03	0.014 460
2^1S	-2.145 678 587 580 5	-2.145 974 046 05	0.013 769
3^1S	-2.060 989 082 34	-2.061 271 989 7	0.013 726
4^1S	-2.033 307 817 48	-2.033 586 717 0	0.013 716
5^1S	-2.020 899 726 1	-2.021 176 851 5	0.013 713
6^1S	-2.014 286 911	-2.014 563 098	0.013 711
7^1S	-2.010 350 14	-2.010 625 77	0.013 710
8^1S	-2.007 818 34	-2.008 093 6	0.013 709
9^1S	-2.006 094 5	-2.006 369 5	0.013 708
10^1S	-2.004 868 1	-2.005 142 9	0.013 706

^a All calculations performed using, $\omega = 25$. For the ground state this corresponds to a 1729×1729 size matrix and for the excited state this corresponds to a 3276×3276 size matrix.

Table 6.4 Energies calculated for the excited triplet states of helium with clamped nucleus, E_∞ and nucleus in motion, E . The helium mass is taken from CODATA 2014^[52], $M = 7294.299\,541\,36$ a.u. Digits listed are converged.

State ^a	E/E_h	E_∞/E_h	$E_{nm}(\%)$
2^3S	-2.174 930 190 712 30	-2.175 229 378 236	0.013 756 1
3^3S	-2.068 405 243 694 68	-2.068 689 067 472	0.013 721 8
4^3S	-2.036 232 827 788	-2.036 512 083 098	0.013 714 2
5^3S	-2.022 341 573 18	-2.022 618 872 302	0.013 711 7
6^3S	-2.015 101 168 23	-2.015 377 452 99	0.013 710 7
7^3S	-2.010 854 227 6	-2.011 129 919 5	0.013 710 1
8^3S	-2.008 151 806	-2.008 427 12	0.013 709 8
9^3S	-2.006 326 454	-2.006 601 516	0.013 709 7
10^3S	-2.005 035 91	-2.005 310 794	0.013 709 6

^a All calculations performed using a 3276×3276 size matrix, $\omega = 25$.

In both the singlet case, Table 6.3 and triplet case, Table 6.4 the energies calculated with nucleus in motion are higher (less negative) than the clamped nucleus case for all ground and excited states, i.e. clamping the nucleus over-estimates the bound state stability. The effect of nuclear motion on the energy, $E_{nm}(\%)$, has a subtly greater effect on the singlet systems than the triplet for the same principle quantum number n . For the 2^1S system, $E_{nm}(\%) = 0.013769 \%$ whereas for 2^3S , $E_{nm}(\%) = 0.013756 \%$ which is only a 0.000013% difference. The 1^1S system shows an increased effect from the motion of the nucleus with a 0.014% change in energy, showing the influence of nuclear motion is greatest when the electron density is closest to the nucleus. As one of the electrons gets excited to a higher principle quantum number n it occupies a more diffuse spatial region, and the other electron will shield it from the nucleus lessening the attraction and the effect of nuclear motion. The percentage change from nuclear motion effects is approximately equal ($E_{nm}(\%) = 0.0137 \%$) for any singlet and triplet electronic excited state from $n = 2 \rightarrow 10$, showing how the shielding effect of the inner electron is strong for any excited S state regardless of the value of n or spin state.

6.3.4 Comparison to Ground State Method

The NLP constraint on the ground state method, $\alpha = \beta$, is a hindrance for excited and weakly bound systems; so a comparison is done to highlight the advantage of allowing the α , β and γ NLPs to be optimised independently. This excited state method is not capable of fully optimising the ground state eigenvalue for singlet two-electron atoms with $Z \geq 2$ due to the ratio $\alpha/\beta \rightarrow 1$ as the electrons are now in the same spatial region causing the hypergeometric functions to require division by 0, equation (6.14). This problem lies in the program rather than the theory, as when $\alpha/\beta \rightarrow 1$ the original ground state method is restored. In the program, as $\alpha/\beta \rightarrow 1$ the required precision of the calculation increases until infinite precision is required when $\alpha = \beta$, which is a numerical issue and not a physical one. Future modifications of this code would see conditions implemented which catch instances where α/β are very small, and handle them separately. Optimising the first excited state, 2^1S , results in good energy values for both the ground state and first excited state. As an example, consider the energies for 1^1S and 2^1S of helium calculated using a matrix size of 816×816

$$1^1S: E_0 = -2.903\,304\,535, \quad E_0^{\text{Lit}} = -2.903\,304\,557 \text{ a.u.}^{[185]}$$

$$2^1S: E_1 = -2.145\,974\,044, \quad E_1^{\text{Lit}} = -2.145\,974\,046 \text{ a.u.}^{[185]}$$

Weakly bound systems such as the hydride anion, H^- can be optimised using the excited state method as α and β are sufficiently unique. The excited state method may calculate an acceptable match for the ground state energy for systems where $Z \geq 2$, however as it cannot currently optimise it directly, it is a calculation best left to the simpler and more efficient ground state method where $\alpha = \beta$ better encapsulates the physics of the system providing a more accurate solution. Electrons in the triplet state have parallel spins meaning that they cannot exist at the same point due to the Pauli exclusion principle making optimisation possible for the triplet ground state 2^3S using the excited state code as the electrons populate different spatial regions meaning α and β can be different.

In order to directly compare between the ground and excited state methods, the matrix size is not a good measure as both methods use different numbering schemes resulting in different matrix sizes for each ω value (Chapter 2 Section 2.2.5). The ground state method can exploit the SYM and ANTISYM numbering schemes, equations (2.23) and (2.24), whereas the excited state method can only use the ASYM numbering scheme, equation (2.22). To overcome this, ω is used to compare both methods as the maximum order of the matrix ($\omega = l + m + n$) is the same for all numbering schemes. The logarithm of the ratio of the calculated energy, E_{calc} with the literature result E_{lit} is determined, equation (6.33), in order to plot all the data points and avoid tabulating them.

$$-\text{Log}\left(\frac{E_{\text{calc}}}{E_{\text{lit}}}\right). \quad (6.33)$$

The expression in equation (6.33) converges to 0 as the accuracy of the calculated energy increases, i.e. as ω increases. The plots in the following sections represent an energy convergence comparison using equation (6.33) between the ground state method using two NLPs (α, γ) and the excited state method using three NLPs (α, β, γ) with crosses (\times) representing the ground state methodology and dots (\bullet) the excited state methodology. All systems were plotted using a clamped nucleus so as to offer a better comparison with the literature values.

6.3.4.1 Singlet state energies

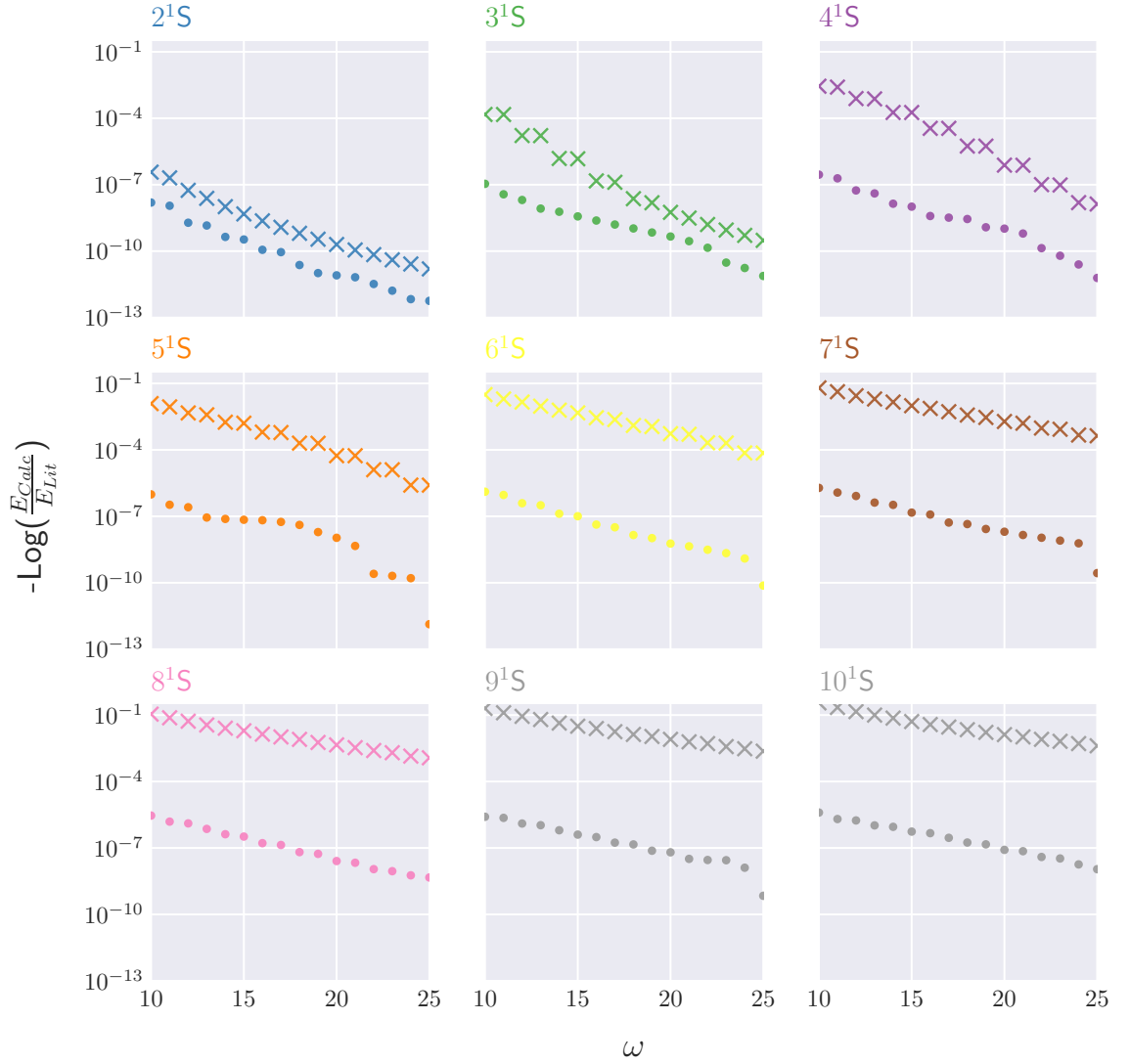


Figure 6.2 Comparison of $-\text{Log}(E_{\text{Calc}}/E_{\text{Lit}})$ with ω for the first 9 excited states of singlet helium with clamped nucleus. For all calculations, E_{Calc} represents the optimised energy value. Crosses (\times) represent the ground state 2 NLP (α, γ) method and dots (\bullet) represent the excited state 3 NLP (α, β, γ) method.

6.3.4.2 Triplet state energies

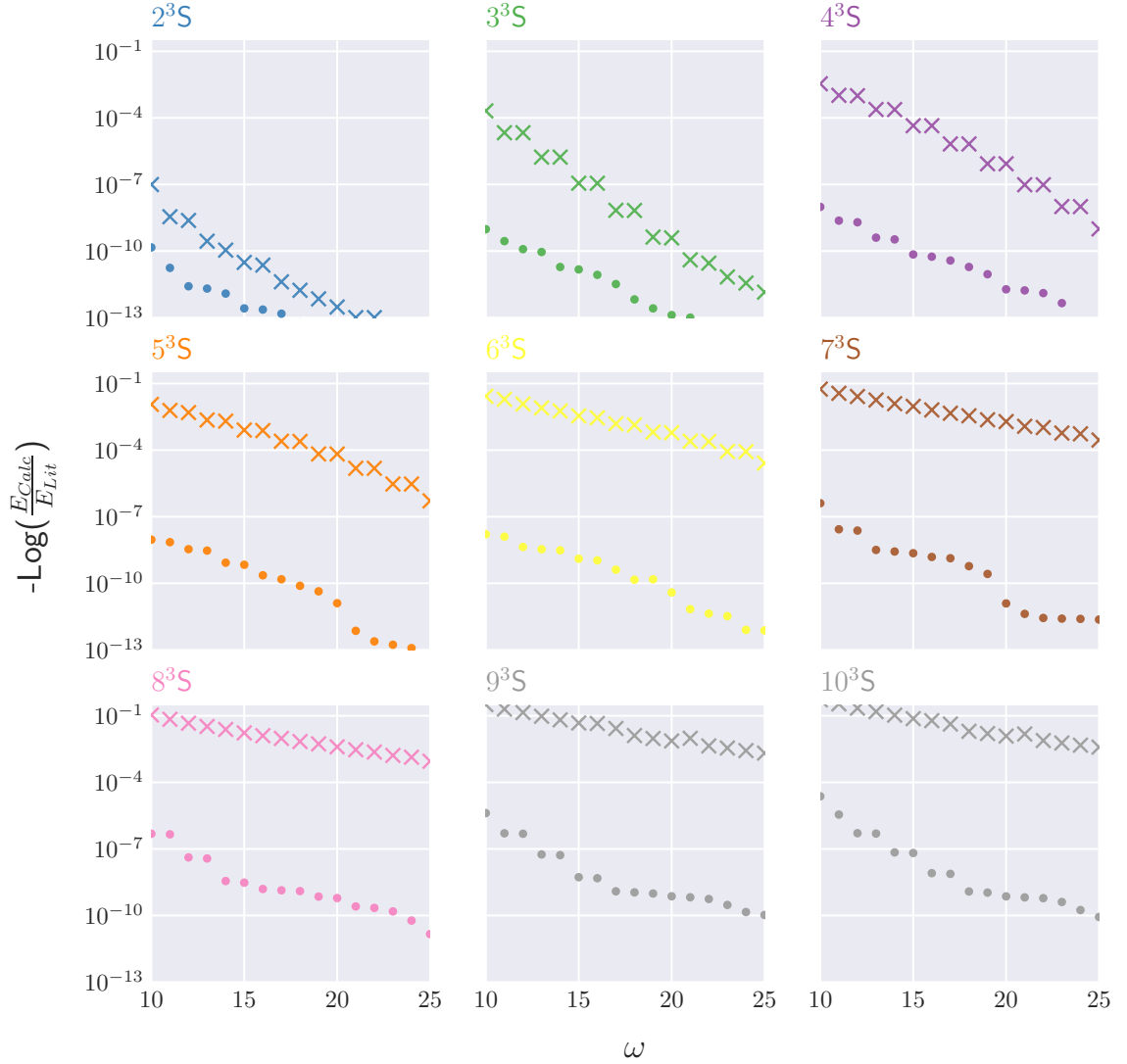


Figure 6.3 Comparison of $-\text{Log}(E_{\text{Calc}}/E_{\text{Lit}})$ with ω for the first 9 excited states of triplet helium with clamped nucleus. For all calculations, E_{Calc} represents the optimised energy value. Crosses (\times) represent the ground state 2 NLP (α, γ) method and dots (\bullet) represent the excited state 3 NLP (α, β, γ) method.

In both Figures 6.2 and 6.3 the excited state method data (\bullet) is below the ground state method data (\times) highlighting a more rapid convergence for all singlet and triplet excited states when using three NLPs (α, β, γ). As you go higher in the excited states, the ground state method is seen to rapidly lose energy convergence characterised by the more horizontal gradient of the data points and an increased separation from

the excited state data sets, whereas the excited state method maintains a strong convergence behaviour. For both singlet and triplet states for $n > 4$ an ω value of 10 (286 size matrix) for the excited state method produces better results than $\omega = 25$ (1729 size matrix) for the ground state method (data points are lower in Figures 6.2 and 6.3). The optimisation for the excited state is less smooth than for the ground state method which is likely due to the extra NLP adding another dimension to the parameter space resulting in a more complicated optimisation calculation. Future work would see an in-depth investigation into the effects of 1,2 and 3 NLPs along with trialling different optimisation algorithms to elucidate how they affect the shape of the PE surface and how effectively the energy minimum is located.

For the lower lying singlet excited states, 2^1S , 3^1S and triplet states 2^3S , 3^3S the ground state method offers a more comparable convergence to the excited state methodology due to the electrons existing closer together and the assumption, $\alpha = \beta$ is still able to encompass the physics of the situation. For higher excited states the ground state methodology rapidly reduces in accuracy to the point where it calculates the 10^1S , 9^3S and 10^3S states as being above the threshold energy, equation (6.31), when $\omega = 30$ equivalent to a 2856 size matrix. The 2^1S and 10^1S convergence data for clamped nucleus helium is listed in Table 6.5 for the ground and excited state method.

Table 6.5 Energy for the 2^1S and 10^1S of clamped nucleus helium calculated using the ground state with two NLPs (α, γ) and the excited state method with three NLPs, α, β, γ . Dashes (-) represent systems where $E > E_{th}$ and n/a for matrix sizes larger than the maximum size generated of 3276×3276 .

ω	2^1S		10^1S	
	α, γ	α, β, γ	α, γ	α, β, γ
5	-2.139 188 945 182	-2.145 956 413 475	-0.544 591 456 459	-
10	-2.145 972 221 491	-2.145 973 968 843	-0.883 304 934 839	-2.005 124 659 196
15	-2.145 974 022 385	-2.145 974 044 392	-1.782 277 094 109	-2.005 140 463 257
20	-2.145 974 045 065	-2.145 974 046 014	-1.945 555 565 169	-2.005 142 612 910
25	-2.145 974 045 976	-2.145 974 046 051	-1.986 311 418 710	-2.005 142 940 887
30	-2.145 974 046 040	n/a	-1.998 809 409 060	n/a
Lit:	-2.145 974 046 054 ^[185]		-2.005 142 991 747 ^[185]	

6.3.5 Expectation values

The energies have been shown to be of high accuracy for the excited states of helium under the clamped nucleus approximation and for the nucleus in motion. The wavefunctions are now tested to assess their quality via four expectation value calculations. The first is to take the inner product of the wavefunction with itself to

check the normalisation to unity, $\langle \Psi_i | \Psi_i \rangle$. The second test is the virial factor, η , the third the nucleus-electron, ν_{31} cusp and the fourth the orthogonality of excited state wavefunctions with lower excited states and the ground state.

Normalisation

The inner product of the excited state wavefunctions is calculated to ascertain its normalisation to unity

$$\langle \Psi_i | \Psi_i \rangle = \int_0^\infty \Psi_i^* \Psi_i dV = 1. \quad (6.34)$$

Virial condition

The virial condition, described in Chapter 2 but reproduced here from equation (2.37). The virial factor, η is used to test the quality of the excited state wavefunctions

$$\eta = \frac{\langle \hat{V} \rangle}{\langle \hat{T} \rangle} + 2 = 0. \quad (6.35)$$

Inter-particle cusps

The nucleus electron, ν_{31} cusp, reproduced here from equation (2.42)

$$\nu_{31} = \langle \hat{\nu}_{31} \rangle = \frac{\langle \Psi | \delta(\mathbf{r}_{31}) \frac{\partial}{\partial r_{31}} | \Psi \rangle}{\langle \Psi | \delta(\mathbf{r}_{31}) | \Psi \rangle}. \quad (6.36)$$

The exact nucleus-electron cusp value for helium with clamped nucleus, $M = \infty$, $Z \equiv Z_3 = 2$, $Z_1 = -1$ is

$$\nu_{31} = Z_3 Z_1 \frac{M m_1}{M + m_1} = Z_3 Z_1 = -2. \quad (6.37)$$

Table 6.6 lists these expectation values calculated for a variety of excited states of helium.

Table 6.6 Normalisation, $\langle\Psi|\Psi\rangle$, virial factor, η and nucleus-electron cusp, ν_{31} calculated for n^1S states, $n = 2, 4, 6, 8, 10$ of helium using a 3276 size matrix and three NLPs. The helium mass is taken from CODATA 2014^[52],
 $M = 7294.299\ 541\ 36$ a.u.

State	$\langle\Psi \Psi\rangle$	η	ν_{31}
2^1S	1.000 000 000 000 002	-3.8×10^{-17}	-1.999 999 999 845
4^1S	1.000 000 000 000 003	-5.2×10^{-17}	-1.999 999 999 618
6^1S	1.000 000 000 000 001	-4.3×10^{-16}	-1.999 999 998 100
8^1S	0.999 999 999 999 997	-2.6×10^{-15}	-1.999 999 979 551
10^1S	1.000 000 000 000 007	-8.8×10^{-15}	-1.999 999 830 177
Exact	1	0	-2

The expectation values in Table 6.6 are of good accuracy for low ($n = 2$) and high ($n = 10$) excited states providing confidence in application of these wavefunctions to explore chemical physics in greater detail.

Orthogonality of excited states

Equation (6.30) shows the orthogonality condition between excited state wavefunctions, and Table 6.7 lists the results for two cases:

1. 10^1S state of helium is optimised and all eigenvectors for lower excited states $n = 1 \dots 9$ are extracted from this single diagonalisation.
2. Each excited state is optimised individually for n^1S for $n = 1 \dots 10$. In the case of $n = 1$ the ground state method is used with equivalent ω .

Table 6.7 Inner product of the excited state 10^1S state of clamped nucleus helium using a 3276 size matrix with lower lying excited states for a single diagonalisation and when each state is optimised individually.

Inner product	Single diagonalisation	Optimisation
$\langle \Psi_{10} \Psi_9 \rangle$	2.05×10^{-13}	0.000 009
$\langle \Psi_{10} \Psi_8 \rangle$	0.71×10^{-13}	0.000 081
$\langle \Psi_{10} \Psi_7 \rangle$	5.58×10^{-13}	0.000 057
$\langle \Psi_{10} \Psi_6 \rangle$	2.61×10^{-13}	0.000 341
$\langle \Psi_{10} \Psi_5 \rangle$	5.16×10^{-13}	0.000 362
$\langle \Psi_{10} \Psi_4 \rangle$	3.02×10^{-13}	0.000 417
$\langle \Psi_{10} \Psi_3 \rangle$	7.10×10^{-13}	0.000 507
$\langle \Psi_{10} \Psi_2 \rangle$	3.75×10^{-12}	0.000 499
$\langle \Psi_{10} \Psi_1 \rangle$	0.50×10^{-13}	0.000 319
$\langle \Psi_{10} \Psi_0 \rangle$	1.28×10^{-13}	0.000 610 ^a

^a Ψ_0 , 1^1S state calculated using ground state method.

From a single diagonalisation, all lower lying excited states including the ground state are orthogonal to 10^1S as seen in Table 6.7 to a high accuracy validating the Hylleraas, Undheim, Macdonald (HUM) theorem^[188,189], but when each state is optimised individually the orthogonality relation is weaker. This is an example for the orthogonality of the 10^1S state of helium, but the same calculations have been done for a large number of other states with the same trend of results obtained.

6.3.6 Critical Nuclear Charge

Following on from previous work^[94], a variational method is used for the direct determination of the threshold nuclear charge for binding, which is referred to as the critical nuclear charge Z_C using the excited state method. Consider the generalised eigenvalue problem

$$\hat{P}\Psi_j = \lambda_j \hat{Q}\Psi_j, \quad j = 0, 1, 2, \dots \quad (6.38)$$

Where \hat{P} is a self-consistent operator bounded from below and \hat{Q} is a positive-definite, self-conjugate operator^[192]. The positive-definite behaviour of \hat{Q} is automatically tested in the Python code by the use of Cholesky decomposition which requires a matrix to be positive definite for a decomposition to be possible. Due to the ill-conditioned nature of the generalised eigenvalue problem for the excited state method,

there are times where \hat{Q} appears to be negative-definite and Cholesky decomposition fails; which is being caused by numerical instability. Increasing the working precision of the Cholesky decomposition calculation shows that \hat{Q} is positive-definite. The lowest eigenvalue, λ_0 in equation (6.38), corresponds to $\min K(\phi)$ where

$$K(\phi) = \frac{\int \phi^* \hat{P} \phi dV}{\int \phi^* \hat{Q} \phi dV}. \quad (6.39)$$

The eigenvalues λ_j are the extremals of K arising from the requirement that the first variation vanishes, i.e. $\delta K \equiv 0$ ^[192]. The Schrödinger equation for two electron atoms, $\{e^-e^-M^{Z+}\}$ where M is the nuclear mass and Z the nuclear charge is

$$\left(-\frac{1}{2}\nabla_1^2 - \frac{1}{2}\nabla_2^2 - \frac{1}{M}\nabla_3^2 - \frac{Z}{r_1} - \frac{Z}{r_2} + \frac{1}{r_{12}}\right)\Psi = E\Psi. \quad (6.40)$$

Scaling the coordinates r_i by Z results in a Z -scaled Schrödinger equation

$$\left(-\frac{1}{2}\nabla_1^2 - \frac{1}{2}\nabla_2^2 - \frac{1}{M}\nabla_3^2 - \frac{1}{r_1} - \frac{1}{r_2} + \frac{1}{Z} \frac{1}{r_{12}}\right)\Psi = \frac{E}{Z^2}\Psi. \quad (6.41)$$

At the stability threshold the wavefunction satisfies equation (6.41) with $E \equiv E_{\text{th}}$ where

$$E \equiv E_{\text{th}} = -\frac{Z^2}{2} \frac{1}{1 + M^{-1}}. \quad (6.42)$$

Where E_{th} represents the ground state hydrogen-like, nucleus-electron system with the second electron at $r = \infty$ and at rest. Substituting equation (6.42) into equation (6.41) gives

$$\left(-\frac{1}{2}\nabla_1^2 - \frac{1}{2}\nabla_2^2 - \frac{1}{M}\nabla_3^2 - \frac{1}{r_1} - \frac{1}{r_2} + \frac{1}{Z} \frac{1}{r_{12}}\right)\Psi = \frac{-\frac{Z^2}{2} \frac{1}{1+M^{-1}}}{Z^2}\Psi. \quad (6.43)$$

Rearranging

$$\left(-\frac{1}{2}\nabla_1^2 - \frac{1}{2}\nabla_2^2 - \frac{1}{M}\nabla_3^2 - \frac{1}{r_1} - \frac{1}{r_2} + \frac{1}{Z} \frac{1}{r_{12}}\right)\Psi = -\frac{1}{2(1 + M^{-1})}\Psi. \quad (6.44)$$

Rearranging to move Z from the LHS of equation (6.44) to the RHS

$$\left(-\frac{1}{2}\nabla_1^2 - \frac{1}{2}\nabla_2^2 - \frac{1}{M}\nabla_3^2 - \frac{1}{r_1} - \frac{1}{r_2} + \frac{1}{2(1 + M^{-1})}\right)\Psi = -\frac{1}{Z} \frac{1}{r_{12}}\Psi. \quad (6.45)$$

At the threshold energy, $E = E_{\text{th}}$, just prior to electron detachment the nuclear charge Z will be at its minimum value, i.e. $Z = Z_C$

$$\underbrace{\left(-\frac{1}{2}\nabla_1^2 - \frac{1}{2}\nabla_2^2 - \frac{1}{M}\nabla_3^2 - \frac{1}{r_1} - \frac{1}{r_2} + \frac{1}{2(1+M^{-1})}\right)}_{\hat{P}} \Psi = \underbrace{-\frac{1}{Z_C}}_{\text{Eigenvalue}} \underbrace{\frac{1}{r_{12}}}_{\hat{Q}} \Psi. \quad (6.46)$$

\hat{P} is bounded from below as a consequence of the standard assumption in the variational principle that the Hamiltonian operator is bounded from below^[54]; and \hat{Q} is a positive-definite, self-conjugate operator. Instead of the energy, the eigenvalue, λ_j , now corresponds to the reciprocal of the threshold charge of the nucleus

$$\lambda_j = -\frac{1}{Z_C}, \quad j \in [0, k], \quad (6.47)$$

where λ_0 represents the ground state eigenvalue and $(\lambda_1, \lambda_2, \dots, \lambda_k)$ excited state eigenvalues and k the maximum eigenvalue.

6.3.6.1 Z_C : Singlet Ground State

As discussed in Section 6.3.4, the excited state method is unable to fully optimise the singlet ground state energy for $Z \geq 2$, but for the system just prior to electron detachment this is not true as the detaching electron will have a larger spatial separation from the nucleus compared to the second electron which remains closer to the nucleus. The excited state wavefunction and method is used to calculate the critical nuclear charge using this variational approach with results for the ground singlet state 1^1S listed in Table 6.8.

Table 6.8 Comparison of the critical nuclear charge, Z_C calculated using the ground state method with two NLPs α, γ and the excited state method using three NLPs, α, β, γ . ω represents the matrix order, bold digits highlight convergence with the literature value and n/a represent matrix sizes that exceed the maximum size generated (3276×3276).

ω	$\alpha, \gamma^{[77]}$		α, β, γ	
	Matrix size	Z_C	Matrix size	Z_C
5	34	0.912 485 462	56	0.911 081 428
10	161	0.911 060 763	286	0.911 030 854
15	444	0.911 030 798	816	0.911 028 351
20	946	0.911 028 526	1771	0.911 028 244
25	1729	0.911 028 276	3276	0.911 028 224 15
30	2856	0.911 028 235	5456	n/a
35	4389	0.911 028 227 1	8436	n/a
40	6391	0.911 028 225 0	12341	n/a
45	8924	0.911 028 224 4	17296	n/a
Lit. Z_C^∞ :		0.911 028 224 077 255 ^[47]		

Table 6.8 shows that the excited state method produces convergence to 9 significant figures with $\omega = 25$ corresponding to a 3276 size matrix, whereas to achieve the same convergence with the ground state method, $\omega = 45$ is required. The excited state method comes with a computational cost requiring larger, dense matrices and a lot of RAM compared to the simpler ground state method but the reward is a better match to the chemical physics and a more compact wavefunction which more accurately fits the true wavefunction.

6.3.6.2 Z_C : Excited Singlet and Triplet States

Calculation of the critical nuclear charge for the singlet ground state was done to assess the improvement that is gained by allowing the NLPs α and β to be independent. The hydride anion, H^- is known to have a single bound state^[193,194] whereas helium has an infinite number of bound states^[195–197] and more generally N -electron systems have an infinite number of bound states if $Z > N - 1$, where Z is the nuclear charge^[197]. As a consequence of this the critical nuclear charge for atomic, two-electron excited singlet or triplet states is theorised to be within the range, $Z = 1, \dots, 2$.

Equation (6.38) represents a generalised eigenvalue problem. In order to calculate the critical nuclear charge for the singlet excited states of helium, the aforementioned

variational method stays the same but instead of the first eigenvalue, the second is extracted representing the first excited singlet state, 2^1S . The Hamiltonian has not changed thus still satisfies the condition on \hat{P} outlined by Rebane and Kuzminskii^[192] along with \hat{Q} remaining positive-definite confirmed via Cholesky decomposition. The second eigenvalue represents λ_1 in equation (6.38) and (6.47), i.e. the first excited state eigenvalue. In order to calculate excited triplet states the sign of the wavefunction in equation (6.9) is changed from $+$ to $-$ as the wavefunction is anti-symmetric to electron exchange and matrices generated for triplet systems.

Table 6.9 shows convergence of the critical nuclear charge for the 2^1S and 2^3S state of helium with clamped nucleus calculated using the excited state wavefunction and critical charge variational method. The optimisation of these systems proved especially complicated and a maximum matrix size of 1771 is presented here due to larger matrices exhibiting numerical instability.

Table 6.9 Critical nuclear charge, Z_C , for the 2^1S and 2^3S state of helium with clamped nucleus.

ω	Matrix Size	Z_C	
		2^1S	2^3S
5	56	-	-
10	286	1.000 001 527	1.000 070 439
15	816	1.000 000 502	1.000 000 490
20	1771	1.000 000 000 1	1.000 000 001
Lit. Charge		1 ^[198]	1 ^[198]

The data in Table 6.9 shows that both the first singlet excited state and ground triplet state converge to a critical nuclear charge of $Z = 1$ which is contrary to the expected value $Z > 1$, although aligns with the definition of the critical nuclear charge having a single bound state, i.e. the ground state. This study is the first variational calculation of these critical charge values however other authors have investigated the critical nuclear charge for excited states, with Baker *et al.* being the first to suggest that the critical nuclear charge for singly excited states was unity^[199] by analysing the convergence behaviour of the $1/Z$ expansion for excited states, although a relatively small number of perturbation coefficients were analysed. Ivanov *et al.*^[200] developed an analytical Rydberg model for triplet excited states applicable to the singular point, $Z_C = 1$ showing that the Rydberg series ceases to exist at $Z = 1$. Katriel *et. al* showed that the critical charge for 2^1S and 2^3S was $Z_C = 1$ by manually calculating the quantum defect for arbitrary charge systems^[198]

noting that at $Z_C = 1$ the quantum defect of the outer electron also equals 1 for S states. Other authors studying the quantum entanglement properties of two electron excited states at the critical charge for point have also used this value for the critical nuclear charge^[198,201,202].

The energy at electron detachment is known, given in equation (6.42), so can be used to calculate the energy at this Z_C value. When $M = \infty$ this gives, $E \equiv E_{\text{th}} = -Z^2/2 = -0.5$. Energies were calculated for 2^1S and 2^3S states of helium using the critical charge values from Table 6.9. For the 1771 size matrix this produced an energy of $E = 0.499\,999\,999\,997$ for 2^1S and $E = 0.499\,999\,999\,973$ for 2^3S in very good agreement with the exact value. This is also the case when a finite nuclear mass, M is used. Bounds had to be placed on the α parameter during the optimisation as it rapidly approached 0 causing the calculation to crash. It was bounded to never go below $\alpha = 1 \times 10^{-10}$ and the final “optimised” value of α was $\alpha = 1 \times 10^{-10}$, showing that unlike in the ground state where a bound state is possible^[47,170,203], the 2^1S and 2^3S systems at the critical nuclear charge are not bound; and form a hydrogen atom with an unbound electron.

NLP values at $Z = Z_C$

The ground state method is unable to be applied to the calculation of the critical nuclear charge for excited states as it offers incredibly poor convergence, even at large matrix sizes seen in Table (6.10).

Table 6.10 Critical nuclear charge, Z_C , for the 2^3S state of helium with clamped nucleus calculated using the ground state method with two NLPs α, γ .

ω	Matrix Size	Z_C	α	γ
30	2856	1.009 6	0.177 211	1.034 784
35	4389	1.007 7	0.163 539	1.035 289
40	6391	1.005 8	0.146 290	1.023 884
45	8924	1.004 9	0.122 630	1.021 774
50	12051	1.004 1	0.118 317	1.023 940
Lit. Charge:		1 ^[198]		

Even with a matrix size of 12051 the ground state method is unable to calculate Z_C accurately, assuming $Z_C = 1$. Table 6.11 shows the results for α, β and Table 6.12 shows the results for α, β, γ both using the excited state method.

Table 6.11 Critical nuclear charge, Z_C , for the 2^1S state of helium with clamped nucleus calculated using the excited state method with two NLPs α, β .

ω	Matrix Size	Z_C	α	β
10	286	1.000 000 843	0.000 000 17	1.000 000 008
15	816	1.000 000 039	1×10^{-9}	1.000 049
20	1771	1.000 000 008	1×10^{-10}	1.000 000 04
Lit. Charge:		1		

Table 6.12 Critical nuclear charge, Z_C , for the 2^1S state of helium with clamped nucleus calculated using the excited state method with three NLPs α, β, γ .

ω	Matrix Size	Z_C	α	β	γ
10	286	1.000 001 527	0.000 038	0.981 800	0.896 383
15	816	1.000 000 015	1×10^{-8}	0.999 707 16	1.000 027 07
20	1771	1.000 000 000 1	1×10^{-10}	0.999 999 14	1.000 000 18
Lit. Charge:		1			

When $Z = Z_C$, $\alpha \rightarrow 0$ for both α, β and α, β, γ showing that the long-range decay of the electron is very slow, implicating that it is unbound from the system. Interesting future work would look at the values of $\langle r_1 \rangle$, $\langle r_2 \rangle$ and $\langle r_{12} \rangle$ just before the critical nuclear charge to assess the bound state behaviour as one of the electrons detaches. Additionally nuclear motion effects could be included to assess the consequence on the calculated critical nuclear charge value.

6.4 Summary and Conclusions

A high accuracy excited state method has been implemented using the work of Pekeris as a foundation^[50] and extending it to include nuclear motion and three Non-Linear variational Parameters (NLPs). The accuracy of the method was assessed via energy convergence testing for both singlet and triplet states, and found to offer excellent energy convergence for excited states with 12 significant figures converged for 2^1S helium and 15 significant figures converged for 2^3S . For the higher excited states 10^1S offered 8 significant figures convergence and 10^3S , 10 significant figures. The effects of nuclear motion on excited states was quantified by calculating the percentage difference in energy with nucleus fixed and in motion; which showed that as the principle quantum number increases the effect of nuclear motion diminishes due to the diffuse nature of the excited electron being shielded from the nucleus by

the other electron. The advantage of allowing, $\alpha = \beta$ in the excited state method was highlighted by comparison to the ground state method using two NLPs, α, γ and exhibited more rapid convergence for all excited states, whilst the ground state method was still able to obtain good results for the lower level excited states.

The wavefunctions for excited states were shown to be of good quality via the calculation of various expectation values including the virial factor, η and nucleus-electron cusp, ν_{31} which were in very good agreement with their exact values. The wavefunctions were found to be automatically orthogonal to lower lying excited states and the ground state for both a single diagonalisation of the secular determinant, along with separate optimisations for each excited or ground state.

The critical nuclear charge, Z_C was calculated to be, $Z_C = 0.911\,028\,224\,15$ using the excited state method for a 3276 size matrix ($\omega = 25$) which converged much faster to the high accuracy literature value^[47] than the ground state method where Z_C was calculated to be $Z_C = 0.911\,028\,224\,4$ using a 8924 size matrix, ($\omega = 45$).

There is a variety of possibilities for future work incorporating further developments of the excited state code including extension to arbitrary precision and generating larger matrices with all parameters left undefined for studying molecular systems. The high accuracy wavefunctions can be used to study the bound state properties of the ground and excited states just before and after their respective critical nuclear charges, Z_C , to look at quasibound behaviour. Reducing the dimensionality of the problem may yield some interesting insights^[204]. Furthermore the critical nuclear charge for both ground and excited states in low dimensionality. The variational method used to calculate, Z_C can be applied to the particle mass^[22] in order to find the critical mass for binding a three-particle system to investigate the stability boundary for excited states of atoms and molecules. All the calculations done in this work have been S state systems ($l = 0$), and the relaxing of the $\alpha = \beta$ parameter described in this chapter is the first step in developing a Laguerre based wavefunction utilising the series solution to accurately describe P states ($l > 0$). This will require substantial work, but will allow for a deeper understanding of the complicated effect of angular momentum on the system. The methods developed here also provide a very accurate means to study quantum entanglement in two electron systems.

CHAPTER 7



Atomic and molecular structure from an all-particle wavefunction

7.1 Introduction

The terms “Atom” and “Molecule” are prevalent throughout physical, chemical and biological sciences forming the foundations of our classical understanding of these fields, to the point where their status is that of an axiom. The IUPAC Compendium of Chemical Terminology defines an atom as^[205]

“Smallest particle still characterising a chemical element. It consists of a nucleus of a positive charge (Z is the proton number and e the elementary charge) carrying almost all its mass (more than 99.9%) and Z electrons determining its size.”

with an example being helium, He, used in this work.

A molecule is defined as^[206]

“An electrically neutral entity consisting of more than one atom, $n > 1$. Rigorously, a molecule, in which $n > 1$ must correspond to a depression on the potential energy surface that is deep enough to confine at least one vibrational state.”

with an example being the hydrogen molecule, H₂.

Adding or removing electrons from molecules results in molecular ions which are defined as^[206]

“An ion formed by the removal from (positive ions) or addition to (negative ions) a molecule of one or more electrons without fragmentation of the molecular structure.”

with an example being the hydrogen molecular ion, H₂⁺, used in this work.

He and H_2^+ are easily categorised by these atom and molecular ion definitions, but more exotic systems such as the positronium negative ion, Ps^- , consisting of two electrons, e^- and a positron, e^+ , $\{e^-e^-e^+\}$ fits neither of them as all masses are equal ($m_{e^-} = m_{e^+} = m_e = 1$ a.u.); meaning a single particle does not carry more than 99.9% of the mass (atom) and it does not consist of more than 1 atom (molecular ion). The binding of electrons to positrons was independently theorised by Wheeler^[207] and Hylleraas^[208], but detected in the laboratory by Mills in 1981^[209]. Standard approximations for systems like Ps^- such as the clamped nucleus approximation for atoms or the Born-Oppenheimer approximation for molecules are not valid.

The current understanding of molecular structure relies on the BO approximation where the structure is identified using the equilibrium structure defined as a local minimum of the potential energy surface^[68,210]. Primas mathematically characterised the Born-Oppenheimer (BO) electronic Schrödinger equation as an asymptotic singular limit of the all-particle case^[211,212]. In 1980, Claverie and Diner suggested that probability density functions calculated using an all-particle wavefunction could be used to identify structural features of molecules^[213]. It is still not clear if classical molecular-structure emerges from quantum theory without the need for additional assumptions or considerations.

Earlier work by Woolley described how molecular structure remains a “*stark mystery which [...] is simply said to be demanded by the known facts*”^[214], detailing how the all-particle Hamiltonian describes the electrons and identical nuclei as indistinguishable particles, whereas electronic structure within the BO approximation does not. Cafiero and Adamowicz studied the H_3^+ molecular ion^[215] calculating the expectation value of the proton-proton distance along with the inter-protonic angle. They discovered that it was not possible to determine whether H_3^+ had a linear or triangular shape due to the indistinguishability of the nuclei; when it is known from experiment that it is an equilateral triangle^[216].

More recently Mátyus *et al.*^[62] discovered that a structural transition is observed when varying the relative masses of the like-charged particles, a_1 of an $a_1^\pm a_1^\pm a_3^\mp$ three-particle system and showed the particle-density profile for Ps^- indicated a molecular-type behaviour. Using this data they predicted that the transition from an atom-like to a molecule-like system occurs within the interval, $0.4 < m_{a_1}/m_{a_3} < 0.8$. They later extracted one- and two- dimensional motifs of classical molecular structure from an all-particle wavefunction using radial and angular densities and a non-Born-Oppenheimer procedure with Gaussian geminal basis functions^[210].

Woolley^[214] discussed how the stationary eigenstates from an all-particle quantum treatment of a system are not interpretable in terms of molecular structure. By the

separation of the nuclear and electronic motions intrinsic to the BO approximation, classical molecular structure may be being introduced as the nuclei are treated as classical particles^[217].

In this chapter the stability of a system is defined as the existence of at least one bound state below the lowest continuum threshold^[54] and has been studied by various authors for Coulombic three-particle systems as a function of mass^[54,172,203,218–222]. King *et al.* calculated the approximate stability domain^[21] and determined the minimum fractional additional binding, g , as a function of the reciprocal mass of the uniquely charged particle, A_3 , in a three-particle system. They suggested that this minimum value of $g(A_3)$ which falls in the interval, m_{a_1}/m_{a_3} , predicted by Mátyus *et al.*, may correspond to the transition from atom-like to molecule-like systems.

Later they calculated the “exact” stability domain using the exact threshold energy^[22] and a variational approach based on the work of Rebane *et al.*^[192].

Chapter summary

The aim of this chapter is to investigate the transition from atom-like systems to molecule-like systems in greater detail, along with the structural features of such systems using a fully correlated method. The first part will replicate and discuss important results from two key papers^[62,21], then go further to try and answer the following questions:

1. *Is a destabilisation in the radial or angular density distributions indicative of an atom-like to molecule-like transition, and if so, is there a definite point at which the transition occurs?*
2. *What is the “geometry” of H^- , Ps^- and H_2^+ ?*

7.2 Method

Centre of mass radial particle densities (COM) were applied to molecular three-particle homonuclear and heteronuclear diatomic systems in Chapter 4 to study the effects of nuclear motion. They are applied in this chapter to a multitude of three-particle systems, so for details on the method and implementation see Chapter 4 Sections 4.2.2 and 4.3. The general form for the angular density is given in equation (2.48) with more details discussed below.

7.2.1 Angular Densities

Written in bra-ket notation, angular densities are calculated using the form in equation (2.49), reproduced here^[210]

$$\Gamma_{a_3, a_1 a_2}(\theta) = \langle \Psi | \delta(\theta_{12} - \theta) | \Psi \rangle, \quad \theta \in [0, \pi]. \quad (7.1)$$

Writing equation (7.1) in its full form gives

$$\Gamma_{a_3, a_1 a_2}(\theta) = \int_0^\infty \int_0^\infty \int_0^\pi \Psi(r_1, r_2, \theta_{12}) \delta(\theta_{12} - \theta) \Psi(r_1, r_2, \theta_{12}) 8\pi^2 r_1^2 r_2^2 \sin(\theta_{12}) d\theta_{12} dr_2 dr_1. \quad (7.2)$$

To obtain this form, the wavefunction needs to be transformed from inter-particle coordinates, r_1, r_2, r_{12} into Eckart coordinates, r_1, r_2, θ_{12} ^[223] shown in Figure 7.1.

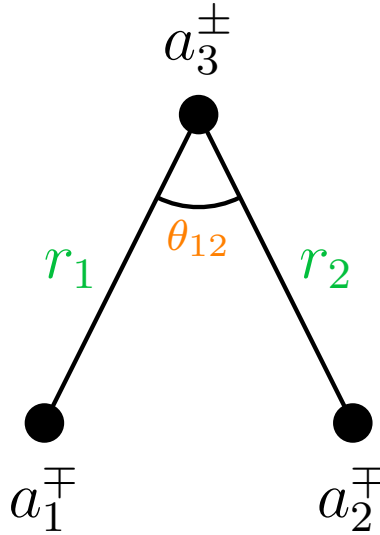


Figure 7.1 Eckart coordinate system, r_1, r_2, θ_{12} .

The coordinate transformation is readily done using the cosine rule. From the dot product

$$r_{12} = \sqrt{r_1^2 + r_2^2 - 2 r_1 r_2 \cos(\theta_{12})}. \quad (7.3)$$

The Jacobian, J , is calculated as

$$J = \left| \frac{\partial(r_1, r_2, r_{12})}{\partial(r_1, r_2, \theta_{12})} \right| = \begin{vmatrix} \frac{\partial r_1}{\partial r_1} & \frac{\partial r_1}{\partial r_2} & \frac{\partial r_1}{\partial \theta_{12}} \\ \frac{\partial r_2}{\partial r_1} & \frac{\partial r_2}{\partial r_2} & \frac{\partial r_2}{\partial \theta_{12}} \\ \frac{\partial r_{12}}{\partial r_1} & \frac{\partial r_{12}}{\partial r_2} & \frac{\partial r_{12}}{\partial \theta_{12}} \end{vmatrix} = \frac{r_1 r_2 \sin(\theta_{12})}{\sqrt{r_1^2 + r_2^2 - 2 r_1 r_2 \cos(\theta_{12})}}. \quad (7.4)$$

Multiplying this by the inter-particle volume element, $dV = 8\pi^2 r_1 r_2 r_{12} dr_1 dr_2 dr_{12}$ ^[12] gives

$$dV = \frac{8\pi^2 r_1^2 r_2^2 r_{12} \sin(\theta_{12})}{\sqrt{r_1^2 + r_2^2 - 2 r_1 r_2 \cos(\theta_{12})}} d\theta_{12} dr_2 dr_1. \quad (7.5)$$

The denominator of equation (7.5) is r_{12} seen in equation (7.3), which conveniently cancels with the r_{12} in the numerator to give

$$dV = 8\pi^2 r_1^2 r_2^2 \sin(\theta_{12}) d\theta_{12} dr_2 dr_1, \quad (7.6)$$

seen in equation (7.2). Angular particle densities were found to be deceptively tricky calculations as when r_{12} in the wavefunction is expressed in terms of θ_{12} *via* equation (7.3), square roots are introduced making integration more difficult, hence cannot be computed in Maple without significant computational overhead. Instead the highly optimised C++ numerical integration codes described in Chapter 3 Section 3.4.3 are used.

7.2.2 Lower bound to stability

It has been proven that all symmetric ($m_{a_1} = m_{a_2}$) systems are stable against dissociation^[224], however no such formal proof exists for unsymmetric systems of the general form $\{m_{a_1}^\pm m_{a_2}^\pm m_{a_3}^\mp\}$ where $m_{a_1} \neq m_{a_2}$, and when the mass-symmetry is broken the stability depends on the relative masses of the particles. Martin *et al.*^[218] discussed the binding of three-unit charges with various constituent masses, and showed that the most convenient variables for the problem are normalised inverse masses

$$A_i = \frac{1/m_{a_i}}{m_{a_1}^{-1} + m_{a_2}^{-1} + m_{a_3}^{-1}}, \quad \text{where } i = 1, 2, 3 \text{ and } A_1 + A_2 + A_3 = 1. \quad (7.7)$$

They predicted the stability domain for unit-charge Coulomb systems and plotted their results on a reciprocal mass ternary diagram. Korobov *et al.*^[225] provided a detailed analysis of the variation of the energy as a function of mass-symmetry breaking in three-particle ions.

King *et al.*^[21] used a series solution method discussed in Chapter 2 to calculate high-accuracy non-relativistic energies and wavefunctions for a wide range of unit-charge, symmetric, three-particle systems of the form $\{m_{a_1}^\pm m_{a_1}^\pm m_{a_3}^\mp\}$ involving a variety of particles: electrons, muons, tauons, protons, deuterons and tritons, and some hypothetical mass systems. The lower bound to stability was calculated from the symmetric wavefunction using the method described in^[218], where the Hamiltonian for an arbitrary system of three particles is written as a sum of symmetric and anti-symmetric terms, i.e., $\hat{H} = \hat{H}_S + \hat{H}_A$, where \hat{H}_S represents terms where $m_{a_1} = m_{a_2}$ and \hat{H}_A represents terms where $m_{a_1} \neq m_{a_2}$.

The variational principle is used to calculate the ground state energy of the asymmetric system using the symmetric ground-state wavefunction, ψ_S of H_S ^[218,54]. The fraction of extra binding, $g(A_3)$, is defined as

$$g(A_3) = \frac{(E_0 - E_{th})}{E_{th}}, \quad (7.8)$$

where A_3 is the reciprocal mass of the uniquely charged particle, equation (7.7), E_0 is the ground-state energy of the symmetric three-body system obtained from the 57-term recursion relation described in Chapter 2, and E_{th} is the two-body threshold energy calculated as

$$E_{th} = E(m_1, m_3) = -\frac{1}{2}\mu_{13} = -\frac{1}{2} \frac{m_1 m_3}{m_1 + m_3}. \quad (7.9)$$

It was found that $g(A_3)$ in equation (7.8) did not decrease monotonically to $A_3 = 0$ (corresponding to an infinite nuclear mass) but contained a minimum which was determined to occur at $A_3 \approx 0.2$ corresponding to the system with a mass ratio of ≈ 0.5 , $(p, p, d)^\pm$. A re-plot of $g(A_3)$ vs. A_3 ^[21] is seen in Figure 7.2.

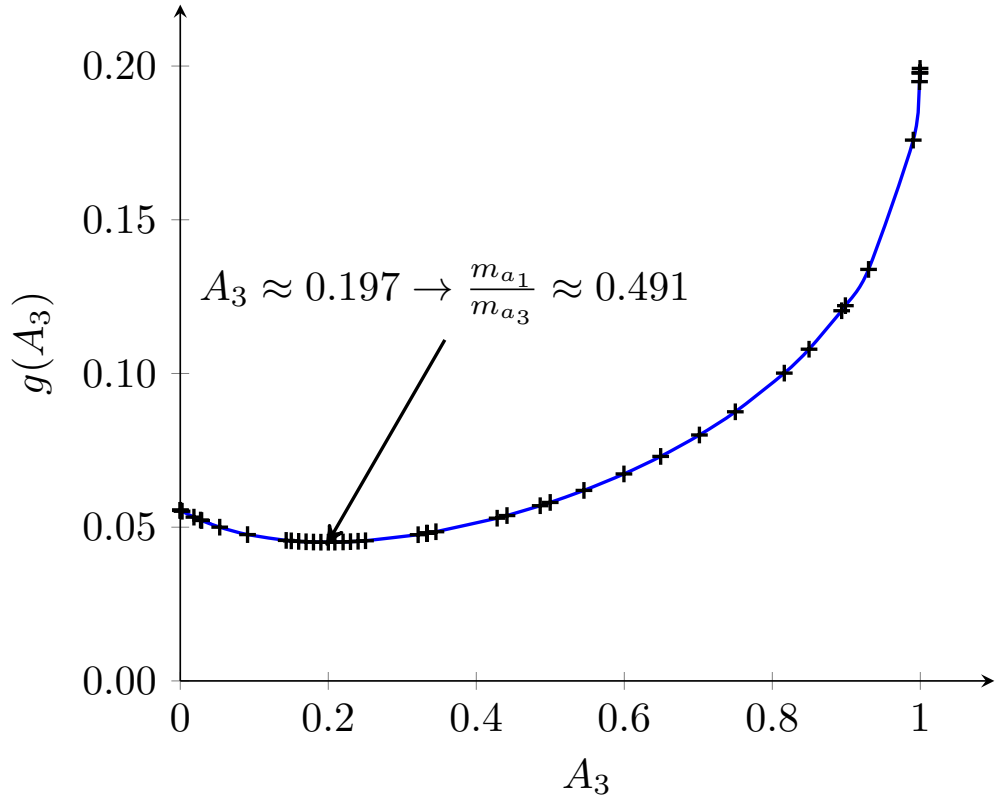


Figure 7.2 Fraction of extra binding, $g(A_3)$ vs. the normalised inverse mass, A_3 , for symmetric systems, $\{m_{a_1}^\pm m_{a_1}^\pm m_{a_3}^\mp\}^{[21]}$. “Reprinted from King *et al.*^[21], with the permission of AIP Publishing.”

The position of this minimum falls in the range suggested by Mátyus for the transition from an atom-like to molecule-like system, i.e. $0.4 < m_{a_1}/m_{a_3} < 0.8$.

7.3 Results and Discussion

All centre of mass and angular particle density distributions in this chapter have been calculated using 2856 size wavefunctions at double precision using the Cuhre algorithm described in Chapter 3 Section 3.4.3.

7.3.1 Emergence of Molecular Structure?

7.3.1.1 Radial densities

The first of the key papers in this discussion is one by Mátyus *et al.*^[62] in which they showed that if the mass ratio of the two identical particles to the third uniquely charged particle mass is small, i.e. $m_{a_3}^\mp \gg m_{a_1}^\pm, m_{a_1}^\pm$, the density of the identical particles is centred at the centre of mass, whereas by increasing the relative mass

ratio, i.e. $m_{a_3}^\mp \ll m_{a_1}^\pm, m_{a_1}^\pm$, a topological change is seen in the particle density and the maxima occurs away from the centre of mass position. This is suggested to correspond to a transition from an atom-like to a molecule-like system in line with results using the Hooke-Calogero model^[226]. Centre of mass radial particle density distributions are calculated in Chapter 4 for molecular systems, and the method developed is re-applied here for a variety of symmetric, unit-charge systems from atoms to molecules and those in-between.

The series solution method used in this work produces more accurate energies and wavefunctions than those of the Gaussian geminal method used by Mátyus *et al.* assessed by energy comparison with literature, $\delta = E_{\text{Lit}} - E$ ^[210], see Table 7.1. The high accuracy literature energy values, E_{Lit} , are taken from Nakashima *et al.*^[28,30] for H^- and H_2^+ and Frolov^[41] for Ps^- , with E representing the calculated energy. The Virial condition for H^- , Ps^- and H_2^+ is also compared in Table 7.1.

Table 7.1 Energy difference with literature, $\delta = E_{\text{Lit}} - E$, and virial factors, η , calculated using the series solution method in this work and the Gaussian geminal method by Mátyus. E_{Lit} are high accuracy energy literature values.

System	δ/E_h		η	
	This work	Mátyus ^[210]	This work	Mátyus ^[210]
H^-	4×10^{-13}	2×10^{-7}	1.7×10^{-12}	-2.6×10^{-7}
Ps^-	1×10^{-13}	9×10^{-8}	1.0×10^{-12}	-8.8×10^{-8}
H_2^+	4×10^{-9}	6×10^{-5}	4.4×10^{-10}	-6.7×10^{-5}

The increased accuracy in this work is beneficial when looking for subtle transition points between atom-like and molecule-like systems.

The centre of mass particle density distributions for an atom, H^- , a molecule, H_2^+ and intermediate system Ps^- are plotted in Figure 7.3.

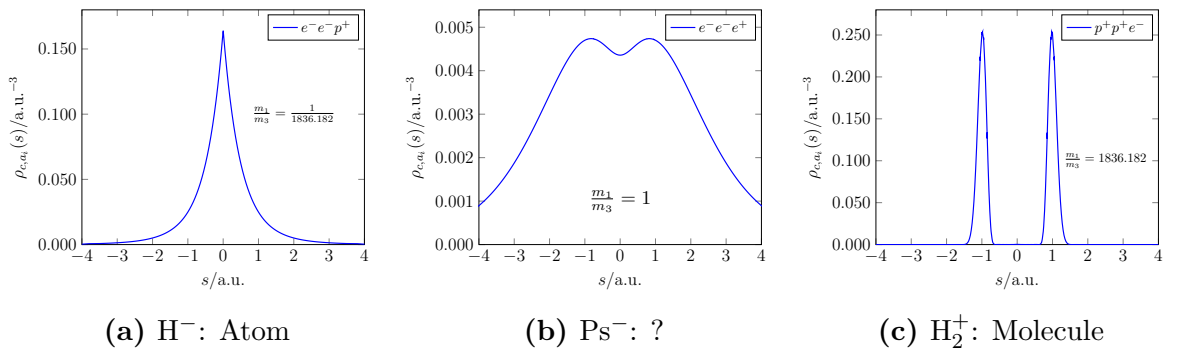


Figure 7.3 Centre of mass particle density distributions for (a) H^- , (b) Ps^- , (c) H_2^+ . The origin represents the centre of mass position in each plot.

The centre of mass particle density distribution for H^- , Figure 7.3 (a) has a maximum at the origin (centre of mass) showing that the greatest particle density is located at the nucleus. Figure 7.3 (c) shows two peaks at a distance of ≈ 1 a.u. from the centre of mass representing the two protons in H_2^+ where the centre of mass sits halfway between them on $r_{12} \equiv r_3$, hence the particle density is greatest around each heavy proton. Figure 7.3 (b) represents Ps^- where all masses are equal, $m_{a_1} = m_{a_2} = m_{a_3} = 1$ a.u., and exhibits an intermediary density distribution shape showing a combination of atom-like and molecule-like behaviour. The maximum in the density distribution is not located at the centre of mass (origin) but instead shows two maxima offset from it similar to H_2^+ , whilst having a high finite particle density at the centre of mass similar to that for H^- . A local minimum in the density is evident at the centre of mass position (origin) which is not present in the other cases.

The mechanism of the transition from Figure 7.3 (a) to (c) is investigated by calculating systems with arbitrary mass ratios as done by Mátyus *et al.*^[62], m_{a_1}/m_{a_3} , where $m_{a_1} = m_{a_2} = 1$ whilst m_{a_3} is varied in order to reproduce their results, but at a greater accuracy, Figure 7.4. For all arbitrary mass systems, a_1 and a_2 have a negative charge, whilst the uniquely charged particle, a_3 , has a positive charge, $\{a_1^- a_2^- a_3^+\}$. This is equivalent to the reverse situation, $\{a_1^+ a_2^+ a_3^-\}$, by charge conjugation invariance. More systems have been included in the range, $0.4 < m_{a_1}/m_{a_3} < 0.8$ hypothesised to be where the atom-like to molecule-like transition occurs^[62].

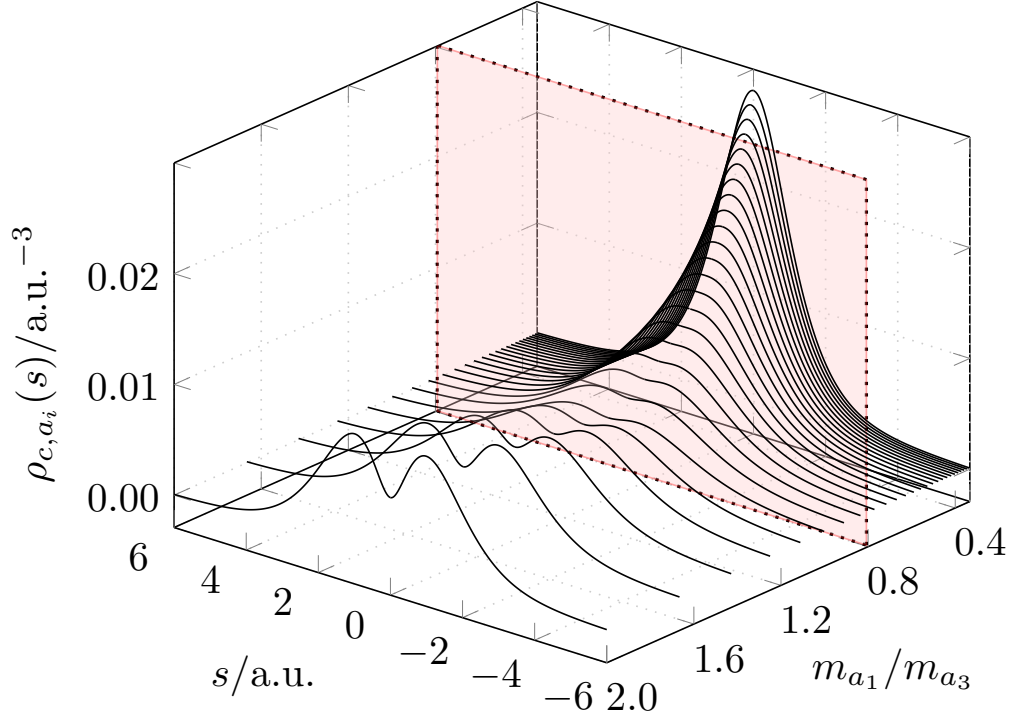


Figure 7.4 Centre of mass particle density distributions for three-particle systems with arbitrary mass ratios, $m_{a_1}/m_{a_3} \in [0.4, 2.0]$. The centre of mass position is located at $s = 0$ a.u. The red shaded plane represents the end point of the transition range predicted by Mátyus, $0.4 < m_{a_1}/m_{a_3} < 0.8$.

The proposed transition range was calculated by analysis of the topology of the centre of mass particle density distributions, but the extent of their analysis is not made clear aside from a visual inspection^[62]. From a visual inspection of Figure 7.4 it too appears to show the transition in the range $m_{a_1}/m_{a_3} \in [0.4, 0.8]$, where at $m_{a_1}/m_{a_3} = 0.4$ there appears to be no minimum at the centre of mass position (origin), and at $m_{a_1}/m_{a_3} = 0.8$ a local minimum can be seen at the centre of mass. A zoomed in view of this region is shown in Figure 7.5.

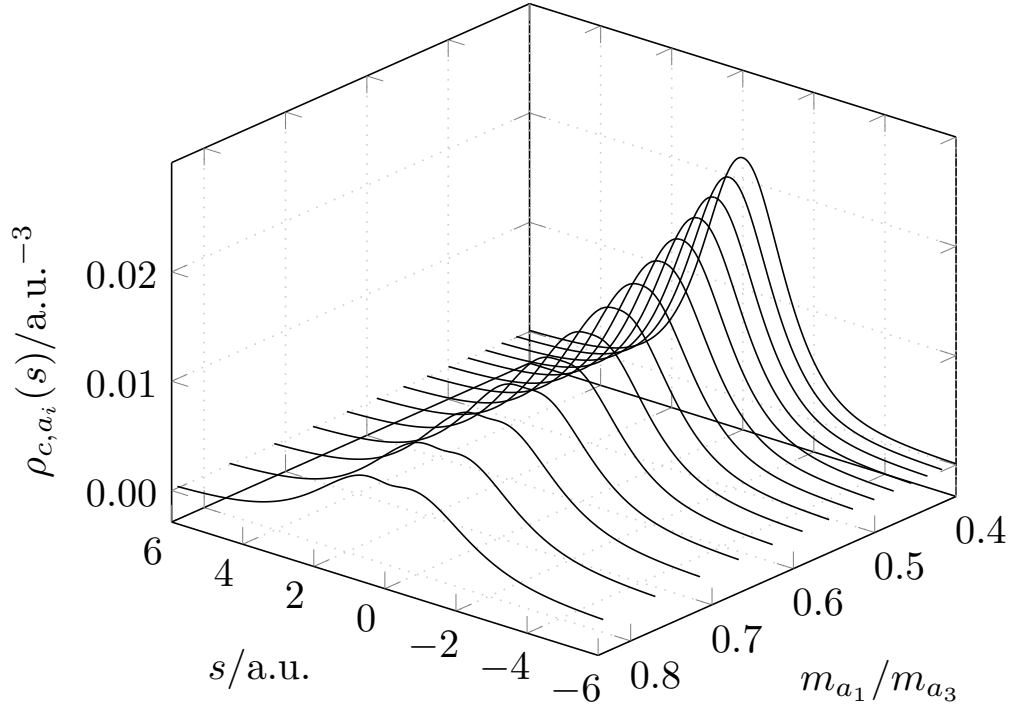


Figure 7.5 Centre of mass particle densities for arbitrary mass ratios, $m_{a_1}/m_{a_3} \in [0.4, 0.8]$. The centre of mass position is located at $s = 0$ a.u.

Figure 7.4 and 7.5 show that by using the high accuracy series solution method and inter-particle coordinates it is possible to replicate the results of Mátyus *et al.*^[62] who use Cartesian coordinates and a Gaussian geminal basis; showing the topological change in the centre of mass particle density distributions moving from an atom-like to molecule-like system.

7.3.1.2 Angular densities

Structural features of atoms and molecules can be analysed using radial densities, but another viewpoint is investigating their angular behaviour with the use of angular densities^[210]. Angular behaviour is not as widely studied in the literature, but several authors have made progress in their understanding for atoms^[168,210,227–231] and molecules^[210].

The angular densities are calculated for θ_{12} seen in Chapter 2 Figure 2.3 using equation (7.2). Figure 7.6 shows angular density plots for, H^- , Ps^- and H_2^+ .

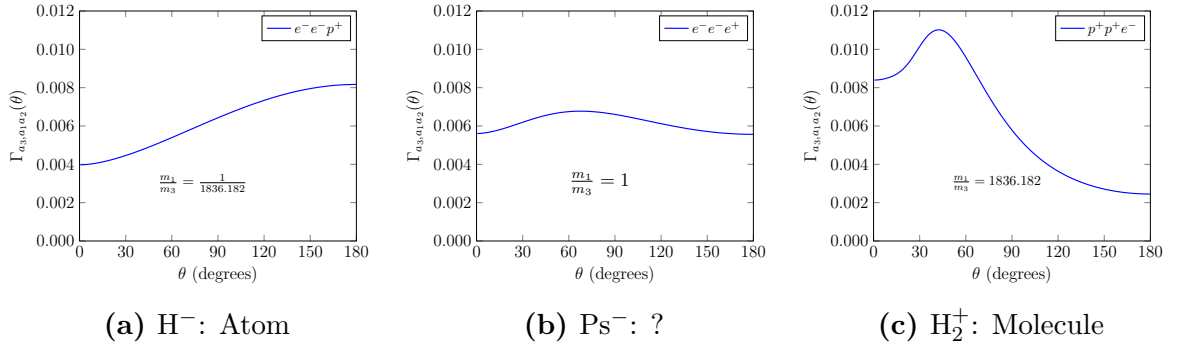


Figure 7.6 Angular particle density distributions for (a) H⁻, (b) Ps⁻, (c) H₂⁺.

Figure 7.6 (a) shows that the most probable angle for H⁻ is $\theta_{12} = 180^\circ$ intuitively showing that the electrons have the highest probability of being located on opposite sides of the nucleus due to their repulsion. The angular density for H₂⁺ in Figure 7.6 (c) is not as intuitive, and has a maximum at $\approx 42^\circ$, whilst Ps⁻ has a small maximum at $\approx 67^\circ$ indicating a delocalised system due to its flat profile, highlighting a weak angular correlation.

Conventional HF theory does not exhibit angular correlation^[210,232] as the two electrons have no preferred angular orientation. The value of this “constant” angular density can be calculated by considering the following normalisation condition

$$8\pi^2 \int_0^\pi \Gamma_{a3,a1,a2}(\theta_{12}) \sin(\theta_{12}) d\theta_{12} = 1, \quad (7.10)$$

and assuming that the angular density is constant, it can be taken outside of the integral

$$8\pi^2 \Gamma_{\text{constant}} \int_0^\pi \sin(\theta_{12}) d\theta_{12} = 1, \quad (7.11)$$

$$\Gamma_{\text{constant}} = \frac{1}{16\pi^2} \approx 0.00663^{[61]}. \quad (7.12)$$

This result has been verified using both the HF code and wavefunctions described in Chapter 5, along with the independent particle model, calculated by removal of $1/r_{12}$ from the Hamiltonian used in the fully correlated method.

For the first time, Figure 7.7 shows the atom-like to molecule-like transition using angular densities calculated for systems with arbitrary mass ratios, m_{a1}/m_{a3} . The plane represents the case of no angular correlation present in HF theory.

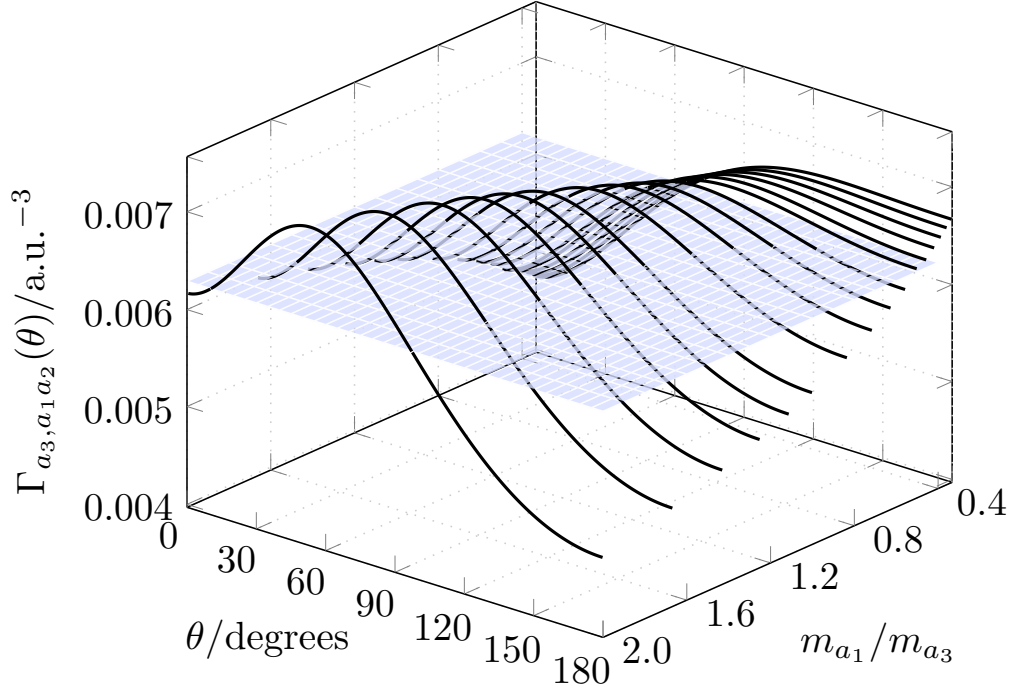


Figure 7.7 Angular particle density distributions for three-particle systems with arbitrary mass ratios, $m_{a_1}/m_{a_3} \in [0.4, 2.0]$. The plane at $\Gamma_{a_3, a_1 a_2}(\theta) = 0.00663$ represents the case of no angular correlation.

Figure 7.7 shows a change in topology in the angular densities. As m_{a_1}/m_{a_3} increases the most probable angle corresponding to the maximum in the density, gradually decreases from 180° to $< 60^\circ$. The constant angular plane is included for a comparison with the case of no angular correlation, acting as a reference point.

In the COM radial particle density transition, Figure 7.4, a local minimum in the density appears at the centre of mass position potentially signalling a transition point; whereas the angular densities do not possess such a clear transition point. Additionally Figure 7.7 highlights how important angular information is lost when the mean field approach is used in HF theory.

7.3.2 Gradient of COM particle density distributions

Figures 7.4 and 7.7 show that by calculating the radial COM and angular particle density distributions, features emerge suggesting that a transition from an atom-like to molecule-like system is occurring. This “transition” has yet to be quantitatively defined and in this work is interpreted as the point at which the gradient in the particle density distributions at the centre of mass position changes sign (inflection point). Atoms exhibit a steep negative gradient in their particle density at the

COM position, Figure 7.3 (a) which gradually increases and eventually changes sign, Figures 7.4 and 7.5.

Firstly the accuracy of the centre of mass particle density distributions need to be assessed. This is achieved by adapting the expression from equation (5.56)^[143]

$$\left(\int_0^\infty [\rho_{c,a_i}^{2856}(s) - \rho_{c,a_i}^{1078}(s)]^2 ds \right)^{1/2}, \quad (7.13)$$

where $\rho_{c,a_i}^{2856}(s)$ and $\rho_{c,a_i}^{1078}(s)$ represent the centre of mass particle density distributions calculated using a 2856 and 1078 size matrix respectively. The maximum error calculated for the centre of mass densities is of the order 4×10^{-8} , and confidence in the gradient calculations is assured at most to this accuracy.

The particle densities in this work are calculated using the sifting theorem of the Dirac delta function described in Chapter 2 Section 2.7, hence are constructed as a series of data points, so the gradient is calculated simply using, $\frac{\Delta \rho_{c,a_i}(s)}{\Delta s}$. These gradient results have been tabulated in Table 7.2 for a variety of mass ratios where $m_{a_1} = m_{a_2} = 1$ and m_{a_3} is varied from 0.1 to 1.8.

Table 7.2 Gradient of the centre of mass particle density distributions at the centre of mass position. Values highlighted in blue represent the mass ratio where the sign change in the gradient occurs.

m_{a_1}/m_{a_3}	$\frac{\Delta \rho_{c,a_i}(s)}{\Delta s}$	m_{a_1}/m_{a_3}	$\frac{\Delta \rho_{c,a_i}(s)}{\Delta s}$
10.00	6.71×10^{-4}	1.000	9.13×10^{-6}
5.000	5.36×10^{-4}	0.909	8.64×10^{-6}
3.333	3.30×10^{-4}	0.833	8.02×10^{-6}
2.500	1.89×10^{-4}	0.769	6.86×10^{-6}
2.000	1.08×10^{-4}	0.714	5.16×10^{-6}
1.666	6.32×10^{-5}	0.666	-2.91×10^{-6}
1.428	3.78×10^{-5}	0.625	-3.05×10^{-6}
1.250	2.30×10^{-5}	0.588	-3.21×10^{-6}
1.111	1.41×10^{-5}	0.555	-7.07×10^{-6}

Table 7.2 shows that the gradient of the particle density at the centre of mass changes sign in the range, $0.666 < m_{a_1}/m_{a_3} < 0.714$, which is within the predicted range $0.4 < m_{a_1}/m_{a_3} < 0.8$ ^[210]. Based on the assumption that the change in sign of the gradient of the COM particle density distributions represents the atom-like to molecule-like transition point, it does not correspond to the system with least

fractional gain in binding energy by association with a third particle^[21] where $m_{a_1}/m_{a_3} \approx 0.491$.

Naturally, the next question to ask is, does $0.666 < m_{a_1}/m_{a_3} < 0.714$ represent the atom-like to molecule-like transition point? To explore this further, the centre of mass particle density is calculated from a_3 to the centre of mass, c , along s_3 shown in Figure 7.8.

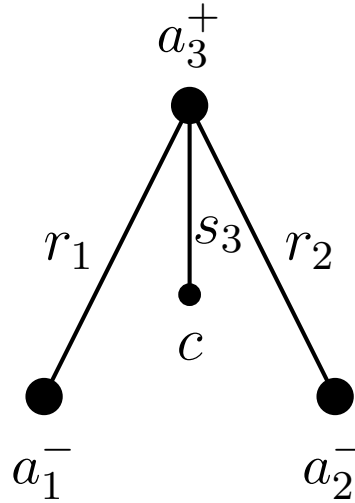


Figure 7.8 Distance, s_3 , from the uniquely charged particle, a_3^+ , to the centre of mass, c .

These centre of mass particle density distributions can be seen in Figure 7.9 for arbitrary mass ratio systems, m_{a_1}/m_{a_3} .

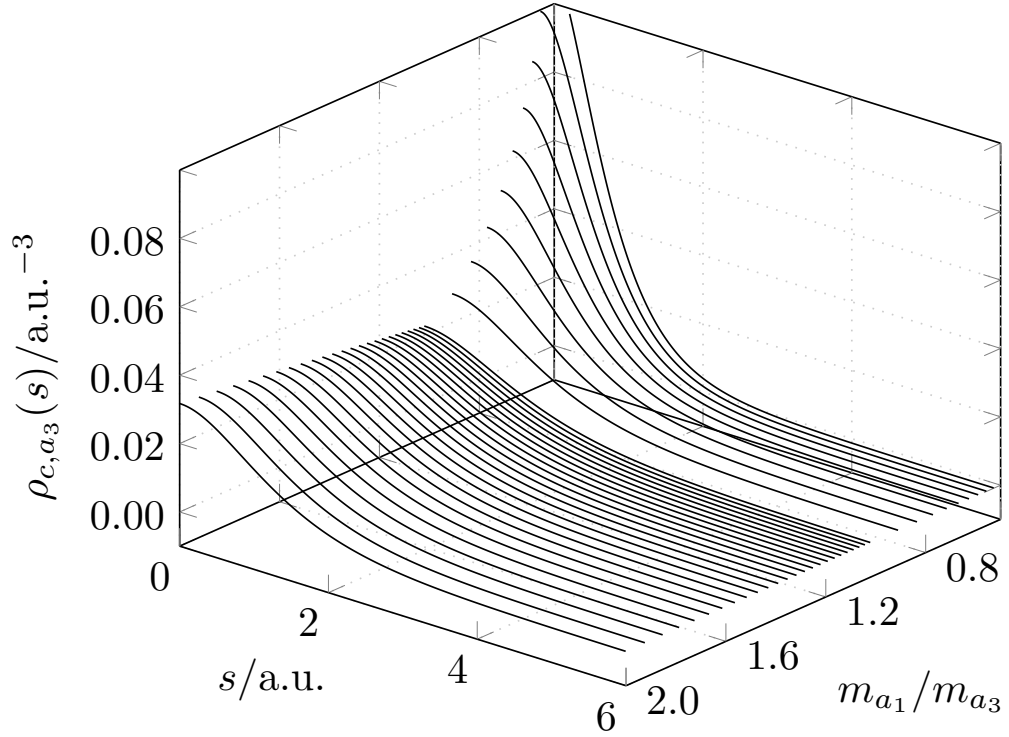


Figure 7.9 Centre of mass particle density distributions for three-particle systems with arbitrary mass ratios, $m_{a_1}/m_{a_3} \in [0.5, 2.0]$ measured from particle a_3 . The centre of mass position is located at $s = 0$ a.u.

If an atom-like to molecule-like transition is described by particle densities or there exists a unique point which defines the transition, it would be expected that other reference points would show a similar topological change describing a change in structure of the system. This is not the case for the centre of mass particle density distribution measured from particle a_3 seen in Figure 7.9 which remains uni-modal for all mass ratios. For low values of m_{a_1}/m_{a_3} the density rapidly increases as the centre of mass position moves closer to the nucleus and closer to an atom-like structure; but as m_{a_1}/m_{a_3} increases towards large values there is no evidence of a bi-modal behaviour or emergence of a maximum away from the centre of mass position (origin), contrary to when it is calculated from particles a_1 and a_2 . As this is measured from one particle, a_3 , rather than both like charged particles, a_1 and a_2 done in Figures 7.4 and 7.5, there is no symmetry about the centre of mass (origin) like in these other Figures.

This shows how the topology of the centre of mass particle density distributions depends on the reference point chosen. Moreso it highlights how care must be taken when using particle densities to explore molecular structure as the reference point selected heavily influences the outcome of the particle density calculation, and important behaviour may be lost if all positions are not considered.

A more complete understanding is required here to assess the usage of particle densities to universally describe molecular structure within the non-Born Oppenheimer situation. However, the analysis here is based on the assumption made in this chapter and the transition point could instead be at the emergence of the two maxima in the particle density distribution or a combination of both behaviours. Future work would see a more broad analysis of the maxima which emerge as m_{a_1}/m_{a_3} increases.

These results also appear to suggest that the transition from atom-like to molecule-like is smooth, so a single definite point cannot be located and quantified. Future work could involve looking at other points within the triangle such as the in-centre, medians, locus etc... and by analysis of the topology of their particle densities, see whether transition points emerge when moving from atom-like to molecule-like systems.

7.3.3 Classical structure from quantum mechanics

The discussion so far has focussed around probability densities in the form of centre of mass and angular particle density distributions. Observables in quantum mechanics are computed using expectation values which represent the average value of the quantity under consideration, see Chapter 2 Section 2.4 for more details. Attempts have been made in this work to reconcile these average values in order to build a geometric picture of three-particle systems.

This is no trivial task as expectation values are not amenable to addition and multiplication, as a function of averages, $f(\langle x \rangle, \langle y \rangle)$ is not the same as the average of the same function, $\langle f(x, y) \rangle$, unless the probability distributions are delta functions^[230].

Figure 7.6 (a) shows the most probable angle for H^- is 180° , yet $\langle r_{12} \rangle \neq \langle r_1 \rangle + \langle r_2 \rangle$. Through extensive testing of geometrical relations, only one was found to hold true from classical geometry; which is the sum of the three internal angles of the triangle, $\Delta a_1^\pm a_2^\pm a_3^\mp$ equals 180° , i.e. $\langle \theta_1 \rangle + \langle \theta_2 \rangle + \langle \theta_{12} \rangle = 180^\circ$, where θ_1 is the angle opposite r_1 and θ_2 opposite r_2 . They were calculated using the following form

$$\langle \theta_i \rangle = \int_0^\infty \int_0^\infty \int_0^\pi \Psi(r_1, r_2, \theta_i) \theta_i \Psi(r_1, r_2, \theta_i) 8\pi^2 r_1^2 r_2^2 \sin(\theta_{12}) d\theta_i dr_2 dr_1, \quad (7.14)$$

where $i = 1, 2$ or 12 . These proved to be very problematic calculations due to complicated and large integrands, which might explain their lack of representation in the literature. Koga has previously calculated $\langle \theta_{12} \rangle$ for two-electron atoms^[227] but the values calculated in Table 7.3 are the first reportings for any electronic and

muonic diatomic molecules. All $\langle\theta_1\rangle$ and $\langle\theta_2\rangle$ calculations for atoms and molecules are the first reportings of these values.

Table 7.3 Expectation values for θ_1, θ_2 and θ_{12} in degrees for three-particle atomic and molecular systems. All values are in degrees.

System	$\langle\theta_1\rangle$	$\langle\theta_2\rangle$	$\langle\theta_{12}\rangle$	$\langle\theta_1\rangle+\langle\theta_2\rangle+\langle\theta_{12}\rangle$
H ⁻	41.448 230 735	41.448 230 735	97.103 538 560	180.000 000 032
He	42.823 840 196	42.823 840 196	94.352 319 628	180.000 000 011
Li ⁺	43.519 344 292	43.519 344 292	92.961 311 544	180.000 000 129
Be ²⁺	43.881 554 689	43.881 554 689	92.236 890 627	180.000 000 006
Ps ⁻	45.632 909 489	45.632 909 489	88.734 181 080	180.000 000 059
T ₂ ⁺	53.434 857 185	53.434 857 185	73.130 287 683	180.000 002 054
D ₂ ⁺	53.400 851 673	53.400 851 673	73.130 287 683	179.999 999 657
H ₂ ⁺	53.323 903 574	53.323 903 574	73.352 192 825	179.999 999 975
Mu ₂ ⁺	52.805 010 616	52.805 010 616	74.389 978 769	180.000 000 002
HD ⁺	53.375 683 726	53.342 521 323	73.281 794 764	179.999 999 814
HT ⁺	53.394 148 881	53.350 026 838	73.255 824 377	180.000 000 097
DT ⁺	53.411 525 810	53.422 478 520	73.165 999 428	180.000 003 758
μ T ₂ ⁺	51.445 227 000	51.445 227 000	77.109 546 031	180.000 000 032
μ D ₂ ⁺	50.991 791 285	50.991 791 285	78.016 417 478	180.000 000 050
μ H ₂ ⁺	50.019 294 463	50.019 294 463	79.961 411 064	179.999 999 992
Mu ^{-a}	45.632 909 489	45.632 909 489	88.734 181 080	180.000 000 059
μ HD ⁺	55.293 198 286	45.635 466 236	79.071 335 478	180.000 000 001
μ HT ⁺	56.939 390 008	44.337 087 955	78.723 522 038	180.000 000 002
μ DT ⁺	52.684 706 757	49.729 595 151	77.585 698 093	180.000 000 003

^a Equivalent to Ps⁻, $m_{a_1} = m_{a_2} = m_{a_3}$.

Table 7.3 shows how this result holds very well to high accuracy for two electron atoms/ions but additionally this is also observed for one electron and one muon molecular systems, even when $m_{a_1} \neq m_{a_2}$. This is a clearly recognisable “classical” angular property which has emerged from the all-particle wavefunction, whilst also acting as a new quality check for the wavefunction itself. The closer the trial wavefunction is to the true wavefunction of the system, the better the agreement of the following relation

$$\langle\theta_1\rangle + \langle\theta_2\rangle + \langle\theta_{12}\rangle = 180^\circ. \quad (7.15)$$

Figure 7.10 shows model diagrams of the most probable and average geometries of H^- , Ps^- and H_2^+ using the most probable and average angles.

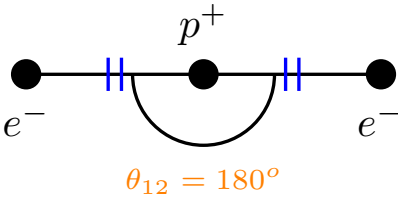
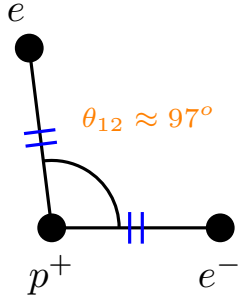
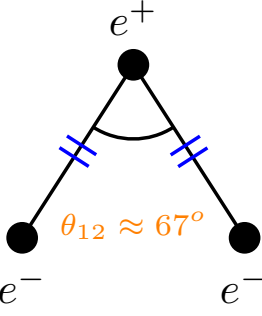
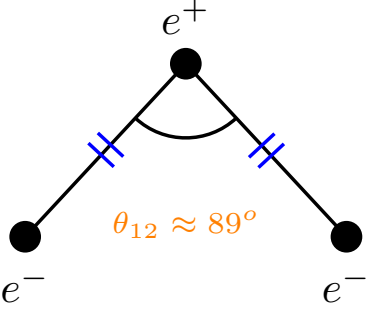
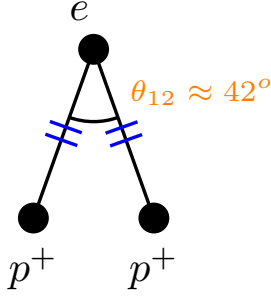
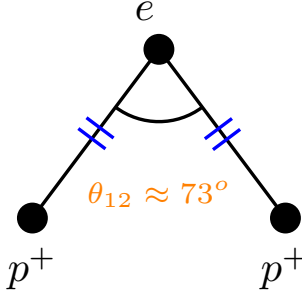
System	Geometry	
	Most probable, θ_{12}	Average, $\langle\theta_{12}\rangle$
H^-		
Ps^-		
H_2^+		

Figure 7.10 Most probable and average features of H^- , Ps^- and H_2^+ . Blue ticks represent congruent sides, i.e. $r_1 = r_2$.

Figure 7.10 shows that Ps^- and H_2^+ both have bent structures for the most probable and average situations, whereas H^- has the highest probability of being linear, but on average is almost a right angled triangle. Additionally, in an angular

sense, the two electrons for H^- are closer in the average situation than the most probable; but for Ps^- they are further apart.

Ps^- and H_2^+ show similar behaviour to one another, with both exhibiting bent structures. Furthermore, for both systems, the average angle is greater than the most probable reflecting the asymmetry in the angular particle densities.

These results suggest that expectation values, alongside particle densities may provide a more complete picture of molecular structure and geometry than particle densities alone; and this investigation is left for interesting future work.

7.4 Summary and Conclusions

Key results were replicated from Mátyus *et al.*^[62] using centre of mass particle densities to study the transition from atom-like to molecule-like systems at greater accuracy whilst using a fully correlated method. This confirmed that a topological transition occurs in the centre of mass particle density distributions as the mass ratio m_{a_1}/m_{a_3} increases for unit-charge, symmetric systems, $a_1^\pm a_1^\pm a_3^\mp$. It has previously been predicted^[62] that the atom-like to molecule-like transition occurs in the range $0.4 < m_{a_1}/m_{a_3} < 0.8$. By assuming that the gradient of the centre of mass particle density distribution at the centre of mass changes sign (stationary point) it was found that the gradient changed sign in the range $0.666 < m_{a_1}/m_{a_3} < 0.714$, within the predicted range. This narrows the atom-like to molecule-like transition range from $0.4 < m_{a_1}/m_{a_3} < 0.8$ ^[62], to $0.666 < m_{a_1}/m_{a_3} < 0.714$ using a more accurate method and wavefunction. Previous work^[21] calculated a lower bound to the stability of three-particle systems whose minimum was hypothesised to represent the atom-like to molecule-like transition and was located in the range predicted by Mátyus for this transition to occur. By the assumption that the stationary point of the density gradient at the centre of mass position dictates the transition point, it was found the minimum of g falls just outside this range, $m_{a_1}/m_{a_3} = 0.491$. Future work would involve going beyond this assumption and analysing further the topological features of the centre of mass particle density distributions at the centre of mass position.

The centre of mass particle density distribution, $\rho_{c,a_3}(s)$, was calculated from the uniquely charged particle, a_3^+ , for arbitrary mass ratios, and no topological transition was found. This shows that the reference point chosen when calculating particle densities will dictate the topology of the resulting density, so when using particle densities to investigate molecular structure, it is important to consider all possible reference points.

For the first time, angular densities have been used to study the atom-like to molecule-like transition using arbitrary mass ratios, m_{a_1}/m_{a_3} . A gradual decrease of the most probable angle, corresponding to the maximum in the density, was found to occur as the ratio m_{a_1}/m_{a_3} increases going from a linear structure in the case of H^- to a bent structure in the case of H_2^+ . Unlike the centre of mass particle densities there was no distinct topological feature representing a shift from atom-like to molecule-like behaviour, instead exhibiting a smooth transition.

Additionally, it was shown how the expectation values of the internal angles of the triangle formed by the three particles, $\Delta a_1^\pm a_2^\pm a_3^\mp$ sum to give 180° , $\langle\theta_1\rangle + \langle\theta_2\rangle + \langle\theta_{12}\rangle = 180^\circ$. This is an example of a classical geometric feature emerging from an all-particle quantum mechanical wavefunction, and acts as a test of the quality of the wavefunction used.

To answer the first initial question posed in this chapter, radial and angular densities do show a topological change as the mass ratio, m_{a_1}/m_{a_3} changes, but the topological behaviour is determined by the reference point chosen. Standalone, particle densities most likely cannot completely describe the molecular structure of a quantum, three-particle system, but certainly complement it as they describe the effective distribution of particles within the system. The particle density was shown to be maximum at the centre of mass for atoms and show a bimodal behaviour with two maxima for molecules; i.e. molecular structure is emerging from an all-particle wavefunction.

To answer the second question posed in this chapter, the hydride anion, H^- , has the highest probability to be linear; but by only calculating the expectation value this information would be lost, as on average it predicts a shape close to that of a right-angled triangle. The positronium negative ion, Ps^- and hydrogen molecular ion, H_2^+ , have the highest probability of exhibiting a bent structure, and on average also exhibit a bent structure.

The work presented in this chapter can be seen as an “exploratory” study into the “geometric” features of three-particle systems, and future work is to be conducted if an answer is desired for whether molecular structure is an intrinsic property rather than a contingent one. Additionally this will provide a more fundamental understanding of two of the most fundamental entities in chemistry and physics, the atom and molecule.

CHAPTER 8



Thesis Summary and Conclusions

The motivation of this work was to study fundamental quantum chemical physics of three-particle atoms and molecules, which has been achieved with explicit inclusion of electron correlation effects and nuclear motion; producing compelling new results into the well established quantum mechanical analogue of the three body problem. Chapter 2 describes the compact series solution method used to solve the time-independent, non relativistic Schrödinger equation to produce high accuracy ground and excited state energies along with corresponding wavefunctions. To achieve these high quality results, a collection of specialised codes written in multiple programming languages have been developed especially for this work to overcome a variety of interesting problems with optimal performance, detailed in Chapter 3.

The Born-Oppenheimer (BO) approximation is a staple of quantum physics and chemistry, but is not assumed in this work. Chapter 4 goes beyond this approximation with explicit inclusion of nuclear motion effects. As the mass of the heavy particles, a_1^+ , a_2^+ , increase the intracule and centre of mass particle density distributions were found to become more localised showing a decrease in the uncertainty in their position. Analysing the centre of mass density distributions for heteronuclear diatomic molecules showed that the nuclear mass ratio is proportional to the maximum probability density and spatial distribution of each particle; a feature which is lost when assuming the Born-Oppenheimer approximation.

The Hartree-Fock (HF) mean field approximation is the second fundamental assumption used in quantum chemistry. Chapter 5 describes a new HF implementation using a Laguerre polynomial basis and, by comparison with the fully-correlated results, highly accurate electron correlation energies were produced for the helium isoelectronic series, even for low nuclear charge Z systems. As the nuclear charge increases, electron correlation diminishes due to the dominance of the nucleus-electron interaction over the electron-electron repulsion. The fully correlated method which explicitly includes r_{12} , demonstrated that the electrons are able to get closer to the nucleus than in the equivalent HF system.

Subtracting the Hartree Fock intracule density from the fully correlated intracule density affirmed the presence of primary and secondary Coulomb holes for helium and the lithium cation and a single Coulomb hole was determined for the hydride anion

with Coulomb hole features quantified. The confirmation of a secondary Coulomb hole for helium and the lithium cation enforces that the effect of electron correlation is still not well understood and more work is needed to decipher its workings. Additionally the critical nuclear charge for binding two electrons to a nucleus was calculated within the HF approximation to be $Z_C^{HF} = 1.031\,177\,528$ which is higher than for the fully correlated system, $Z_C^{FC} = 0.911\,028\,224\,077\,255\,73$ ^[47] as a consequence of a lack of Coulomb electron correlation. The radial behaviour near Z_C^{HF} was investigated using the expectation values of the inter-particle distances r_1 and r_{12} with results implying that the nucleus-electron and electron-electron distances vary smoothly as the nuclear charge decreases, indicating the electrons remain quasi-bound for $Z > 0.85$ within the mean-field potential.

The wavefunction used for all ground state calculations in this work has the variational parameter constraint, $\alpha = \beta$ for systems containing two identical fermions. This enforces the Laguerre orthogonality condition which results in the formation of a sparse secular determinant. Chapter 6 adapts this ground state methodology to excited S states using the work of Pekeris^[50] as a foundation, but extending it for finite nuclear mass and 3 NLPs where all 3, α, β, γ can be unique. This method was found to be computationally demanding, but a matrix size of 3276 was used to produce high accuracy energy eigenvalues for n^1S and n^3S states of helium, $n = 2, \dots, 12$ along with high accuracy wavefunctions tested using virial and cusp conditions. The effect of nuclear motion on excited states was quantified by studying the difference in energy between the moving nucleus and clamped nucleus cases. The energy difference was found to be almost constant for all single and triplet excited states $n^1S, n^3S, n > 1$ highlighting the screening effect of the inner electron is significant for all the excited states considered. The critical nuclear charge to bind two electrons to a nucleus in the ground state was calculated using this more flexible wavefunction and found to converge faster to the true value^[47] than the original ground state method, reinforcing the importance of allowing α and β to be independent. Additionally, the critical nuclear charge for the first excited singlet state, 2^1S and triplet ground state, 3^1S was calculated variationally for the first time and found to converge to unity for both systems.

The notion of atoms and molecules is so ingrained into physical definition that they are treated as axioms. Molecular structure is only understood when assuming the Born-Oppenheimer approximation and Chapter 7 investigates how one transitions from an atom to a molecule. Previous calculations were replicated with a greater accuracy and it was shown that a topological transition occurs in the radial centre of mass particle densities as the mass ratio, m_{a1}/m_{a3} changes confirming previous

investigations^[62]. Using a more accurate method and wavefunction in this thesis, this transition range has been narrowed to $0.666 < m_{a_1}/m_{a_3} < 0.714$. For the first time, angular densities were used to study the atom-like to molecule-like transition, and it was found that there was no clear topological feature which signified the transition, unlike in the radial centre of mass density case. It was also shown how a classical angular property, the sum of the internal angles of a triangle total 180° , emerges from an all-particle quantum mechanical wavefunction.

The work presented in this thesis shows that interesting quantum chemical effects are lost when assuming the Born Oppenheimer and mean field approximations. With the advancement in computational technology it is becoming possible to more accurately model quantum systems and the results shown in this work offer an insight into the simplest, complex problem in quantum physics and chemistry; which will prove useful for future applications and for developing more accurate theories.



Bibliography

- [1] I. Newton, *Philosophiae naturalis principia mathematica*. J. Societatis Regiae ac Typis J. Streater, 1687.
- [2] W. D. Macmillan, "On Poincaré's correction to Bruns' theorem," *B. Am. Math. Soc.*, vol. 19, pp. 349–355, 1913.
- [3] K. Sundman, "Mémoire sur le problème des trois corps," *Acta. Math.*, vol. 36, pp. 105–179, 1912.
- [4] Q. D. Wang, "The global solution of the n-body problem," *Celest. Mech. and Dyn. Astr.*, vol. 26, 1991.
- [5] F. Diacu, "The Solution of the n-body Problem," *Math. Intelligencer.*, vol. 18, pp. 66–70, 1996.
- [6] Ø. Burrau, "Berechnung des Energiewertes des Wasserstoffmolekel-Ions (H_2^+) im Normalzustand," *Naturwissenschaften.*, vol. 15, pp. 16–17, 1927.
- [7] E. A. Hylleraas, "Neue berechnung der energie des heliums im grundzustande, sowie des tiefsten terms von ortho-helium," *Z. Physik.*, vol. 54, pp. 347–366, 1929.
- [8] J. Bartlett, J. Gibbons, and C. Dunn, "The Normal Helium Atom," *Phys. Rev.*, vol. 47, pp. 679–680, 1935.
- [9] J. Bartlett, "The helium wave equation," *Phys. Rev.*, vol. 51, pp. 0661–0669, 1937.
- [10] V. Fock, "On the Schrödinger equation of the helium atom I," *Kong. Norsk. Vidensk.*, vol. 31, pp. 138–144, 1958.
- [11] J. D. Morgan III, "Convergence properties of Fock's expansion for S-state eigenfunctions of the helium atom," *Theor. Chim. Acta.*, vol. 69, pp. 181–223, 1986.
- [12] C. L. Pekeris, "Ground state of two-electron atoms," *Phys. Rev.*, vol. 112, pp. 1649–1658, 1958.
- [13] A. S. Coolidge and H. M. James, "On the Convergence of the Hylleraas Variational Method," *Phys. Rev.*, vol. 51, pp. 855–859, 1937.
- [14] A. A. Frost, "Approximate Series Solutions of Nonseparable Schrödinger Equations. I. Mathematical Formulation," *J. Chem. Phys.*, vol. 41, pp. 478–482, 1964.
- [15] A. A. Frost, M. Inokuti, and J. P. Lowe, "Approximate Series Solutions of Nonseparable Schrödinger Equations. II. General Three-Particle System with Coulomb Interaction," *J. Chem. Phys.*, vol. 41, pp. 482–489, 1964.

-
- [16] H. Cox, S. Smith, and B. Sutcliffe, "Some calculations on the ground and lowest-triplet state of helium in the fixed-nucleus approximation," *Phys. Rev. A.*, vol. 49, pp. 4520–4532, 1994.
- [17] H. Cox, S. Smith, and B. Sutcliffe, "Some calculations on the ground and lowest-triplet state of the helium isoelectronic sequence with the nucleus in motion," *Phys. Rev. A.*, vol. 49, pp. 4533–4539, 1994.
- [18] H. Cox, P. Sinclair, S. Smith, and B. Sutcliffe, "Some calculations on the ground 1S state of the positronium negative ion," *Mol. Phys.*, vol. 87, pp. 399–406, 1996.
- [19] F. Saavedra, E. Buendía, F. Gálvez, and A. Sarsa, "Precise variational calculation of some s-states of coulomb three-body systems with two identical particles," *Eur. Phys. J. D*, vol. 2, pp. 181–190, 1998.
- [20] F. Saavedra, E. Buendía, F. Gálvez, and A. Sarsa, "Variational calculation of some s-states of coulomb three-body systems," *Eur. Phys. J. D*, vol. 13, pp. 201–206, 2001.
- [21] A. King, F. Longford, and H. Cox, "The stability of s-states of unit-charge coulomb three-body systems," *J. Chem. Phys.*, vol. 139, pp. 224306–1–224306–7, 2013.
- [22] A. King, P. E. Herlihy, and H. Cox, "Predicting the stability of atom-like and molecule-like unit-charge coulomb three-particle systems," *J. Chem. Phys.*, vol. 141, pp. 044120–044120, 2014.
- [23] A. L. Baskerville, A. W. King, and H. Cox, "Quantum effects of nuclear motion in three-particle diatomic ions," *Phys. Rev. A.*, vol. 94, pp. 042512–1–042512–9, 2016.
- [24] A. M. Frolov, "Bound-state properties of negatively charged hydrogenlike ions," *Phys. Rev. A.*, vol. 58, pp. 4479–4483, 1998.
- [25] A. M. Frolov, "Two-stage strategy for high-precision variational calculations," *Phys. Rev. A.*, vol. 57, pp. 2436–2439, 1998.
- [26] A. M. Frolov, "Field shifts and lowest order QED corrections for the ground 1 1S and 2 3S states of the helium atoms," *J. Chem. Phys.*, vol. 126, pp. 104302–104302, 2007.
- [27] A. M. Frolov, "On the bound states in the muonic molecular ions," *Eur. Phys. J. D.*, vol. 66, pp. 1–11, 2012.
- [28] H. Nakashima and H. Nakatsuji, "Solving the schrödinger equation for helium atom and its isoelectronic ions with the free iterative complement interaction (ici) method," *J. Chem. Phys.*, vol. 127, pp. 224104–1–224104–14, 2007.
- [29] H. Nakashima and H. Nakatsuji, "Solving the electron-nuclear schrödinger equation of helium atom and its isoelectronic ions with the free iterative-complement-interaction method," *J. Chem. Phys.*, vol. 128, pp. 154107–1–154107–7, 2008.

-
- [30] Y. Hijikata, H. Nakashima, and H. Nakatsuji, “Solving non-born–oppenheimer schrödinger equation for hydrogen molecular ion and its isotopomers using the free complement method,” *J. Chem. Phys.*, vol. 130, pp. 024102–1–024102–11, 2009.
- [31] P. J. Mohr, B. N. Taylor, and D. B. Newell, “Codata recommended values of the fundamental physical constants: 2006,” *Rev. Mod. Phys.*, vol. 80, pp. 633–730, 2008.
- [32] M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, B. Mennucci, G. A. Petersson, H. Nakatsuji, M. Caricato, X. Li, H. P. Hratchian, A. F. Izmaylov, J. Bloino, G. Zheng, J. L. Sonnenberg, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, J. A. Montgomery, Jr., J. E. Peralta, F. Ogliaro, M. Bearpark, J. J. Heyd, E. Brothers, K. N. Kudin, V. N. Staroverov, R. Kobayashi, J. Normand, K. Raghavachari, A. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, N. Rega, J. M. Millam, M. Klene, J. E. Knox, J. B. Cross, V. Bakken, C. Adamo, J. Jaramillo, R. Gomperts, R. E. Stratmann, O. Yazyev, A. J. Austin, R. Cammi, C. Pomelli, J. W. Ochterski, R. L. Martin, K. Morokuma, V. G. Zakrzewski, G. A. Voth, P. Salvador, J. J. Dannenberg, S. Dapprich, A. D. Daniels, O. Farkas, J. B. Foresman, J. V. Ortiz, J. Cioslowski, and D. J. Fox, “Gaussian 09 Revision D.01.” Gaussian Inc. Wallingford CT 2009.
- [33] G. te Velde, F. M. Bickelhaupt, E. J. Baerends, C. Fonseca Guerra, S. J. A. van Gisbergen, J. G. Snijders, and T. Ziegler, “Chemistry with adf,” *J. Comput. Chem.*, vol. 22, pp. 931–967, 2001.
- [34] A. M. Frolov, “Bound-state properties and muon sticking probabilities for the $s(l = 0)$ -states in non-symmetric muonic molecular ions,” *J. Phys. B*, vol. 34, pp. 3813–3828, 2001.
- [35] A. M. Frolov, “Structures and properties of the ground states in h_2^+ -like adiabatic ions,” *J. Phys. B*, vol. 35, p. L331, 2002.
- [36] A. M. Frolov and V. Smith, “Bound state properties and astrophysical applications of negatively charged hydrogen ions,” *J. Chem. Phys.*, vol. 119, pp. 3130–3137, 2003.
- [37] A. M. Frolov, “Atomic and molecular three-body ions with positively charged muons,” *Phys. Rev. A*, vol. 69, p. 022505, 2004.
- [38] A. M. Frolov, “Isotopic effects for the ground $1s1(l=0)$ states in the light two-electron ions,” *J. Chem. Phys.*, vol. 124, pp. 224323–1–224323–10, 2006.
- [39] A. M. Frolov, “Field shifts and lowest order QED corrections for the ground $1\ 1S$ and $2\ 3S$ states of the helium atoms,” *J. Chem. Phys.*, vol. 126, pp. 104302–104302, 2007.
- [40] A. M. Frolov and D. Wardlaw, “Bound state spectra of three-body muonic molecular ions,” *Eur. Phys. J. D*, vol. 63, pp. 339–350, 2011.
- [41] A. M. Frolov, “Annihilation, bound state properties and photodetachment of the positronium negatively charged ion,” *Chem. Phys. Lett*, vol. 626, pp. 49–54, 2015.

-
- [42] G. W. F. Drake and Z.-C. Yan, “Energies and relativistic corrections for the rydberg states of helium: Variational results and asymptotic analysis,” *Phys. Rev. A.*, vol. 46, p. 2378, 1992.
 - [43] S. P. Goldman and G. W. F. Drake, “Two-electron lamb shifts and $1s2s\ 3s1-1s2p\ 3pj$ transition frequencies in helium-like ions,” *J. Phys. B.*, vol. 17, p. L197, 1984.
 - [44] G. W. F. Drake, “Theory of Relativistic Magnetic Dipole Transitions: Lifetime of the Metastable 2^S3 State of the Heliumlike Ions,” *Phys. Rev. A.*, vol. 3, p. 908, 1971.
 - [45] Z.-C. Yan and G. W. F. Drake, “High-precision calculations of the zeeman effect in the $2\ 3pj$, $2\ 1p1$, $2\ 3s1$, and $3\ 3pj$ states of helium,” *Phys. Rev. A.*, vol. 50, p. R1980, 1994.
 - [46] G. W. F. Drake, “High-Precision Variational Calculations for the $1s21s$ State of H^- and the $1s21s, 1s2s1s$ and $1s2s3s$ States of Helium,” *Nuclear Instruments & Methods in Physics Research Section B-Beam Interactions with Materials and Atoms*, vol. 31, pp. 7–13, 1988.
 - [47] C. Estienne, M. Busuttill, A. Moini, and G. W. F. Drake, “Critical nuclear charge for two-electron atoms,” *Phys. Rev. Lett.*, vol. 112, pp. 173001–173001–5, 2014.
 - [48] T. P. Softley, “Applications of molecular Rydberg states in chemical dynamics and spectroscopy,” *International Reviews in Physical Chemistry*, vol. 23, pp. 1–78, 2004.
 - [49] M. Saffman, T. G. Walker, and K. Mølmer, “Quantum information with Rydberg atoms,” *Reviews of Modern Physics*, vol. 82, pp. 2313–2363, 2010.
 - [50] C. L. Pekeris, “Excited S States of Helium,” *Phys. Rev.*, vol. 127, pp. 509–511, 1962.
 - [51] P. J. Mohr, B. N. Taylor, and D. B. Newell, “Codata recommended values of the fundamental physical constants: 2010,” *Rev. Mod. Phys.*, vol. 84, pp. 1527–1605, 2012.
 - [52] P. J. Mohr, B. N. Taylor, and D. B. Newell, “The 2014 codata recommended values of the fundamental physical constants (web version 7.1), <http://physics.nist.gov/constants>,” 2015.
 - [53] H. Bethe and E. Salpeter, *Quantum mechanics of one and two electron atoms*. Dover Publications, 1957.
 - [54] E. A. G. Armour, J.-M. Richard, and K. Varga, “Stability of few-charge systems in quantum mechanics,” *Phys. Rep.*, vol. 413, pp. 1–90, 2005.
 - [55] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions*. with Formulas, Graphs, and Mathematical Tables, Courier Corporation, 2012.
 - [56] F. Mandl, *Quantum Mechanics*. Wiley, 1992.
 - [57] T. Kato, “On the eigenfunctions of many-particle systems in quantum mechanics,” *Commun. Pure Appl. Math.*, vol. 10, pp. 151–177, 1957.

-
- [58] R. T. Pack and W. B. Brown, "Cusp Conditions for Molecular Wavefunctions," *J. Chem. Phys.*, vol. 45, pp. 556–559, 1966.
- [59] W. Bingel, "The behaviour of the second-order density matrix at the coulomb singularities of the schrödinger equation," *Theor. Chim. Acta.*, vol. 5, pp. 341–345, 1966.
- [60] Y. K. Ho, "Positron annihilation in the positronium negative ion," *J. Phys. B.*, vol. 16, pp. 1503–1509, 1983.
- [61] E. Mátyus, J. Hutter, U. Müller-Herold, and M. Reiher, "Extracting elements of molecular structure from the all-particle wave function," *J. Chem. Phys.*, vol. 135, p. 204302, 2011.
- [62] E. Mátyus, J. Hutter, U. Müller-Herold, and M. Reiher, "On the emergence of molecular structure," *Phys. Rev. A.*, vol. 83, pp. 052512–1–052512–5, 2011.
- [63] P. Dirac, *The Principles of Quantum Mechanics*. Oxford: Oxford University Press, 1981.
- [64] P. Dirac, "The principles of quantum mechanics," 1947.
- [65] M. Born and R. Oppenheimer, "Zur quantentheorie der molekeln," *Ann. Phys.*, vol. 84, pp. 457–484, 1927.
- [66] P. Atkins and J. De Paula, *Atkins' Physical Chemistry*. Oxford: Oxford University Press, 8 ed., 2006.
- [67] D. R. Hartree, "The Wave Mechanics of an Atom with a non-Coulomb Central Field. Part III. Term Values and Intensities in Series in Optical Spectra," in *Proceedings of the Cambridge Philosophical Society*, p. 426, 1928.
- [68] E. Lewars, *Computational Chemistry*. Introduction to the Theory and Applications of Molecular and Quantum Mechanics, Springer Science & Business Media, 2010.
- [69] B. Stroustrup, *The C++ programming language*. Addison-Wesley, 1986.
- [70] D. Redfern, *The Maple handbook*. Maple V release 3, Springer, 1994.
- [71] R. V. Guido, "The python language reference, python software foundation:." <http://docs.python.org/py3k/reference/index.html>.
- [72] E. Ziegel, W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes: The Art of Scientific Computing*. Technometrics, 1987.
- [73] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide*. Philadelphia, PA: Society for Industrial and Applied Mathematics, third ed., 1999.
- [74] C. L. Lawson, R. J. Hanson, D. R. Kincaid, and F. T. Krogh, "Basic linear algebra subprograms for fortran usage," *ACM Trans. Math. Softw.*, vol. 5, pp. 308–323, 1979.

-
- [75] B. Schaling, *The Boost C++ Libraries*. Laguna Woods: XML Press, 2 ed., 2014.
 - [76] “Exprrtk.” <http://www.partow.net/programming/exprrtk/index.html>. Accessed: 2017-07-07.
 - [77] A. King, *The Stability of Small Atoms and Molecules: A Quantum Mechanical Three-Body Study*. PhD thesis, University of Sussex, 2015.
 - [78] G. W. F. Drake and Z. Yan, “Variational eigenvalues for the S states of helium,” *Chem. Phys. Lett.*, vol. 229, pp. 486–490, 1994.
 - [79] P. Holoborodko, “MPFR C++,” holoborodko.com/pavel/mpfr/, p. GNU General Public License, 2008.
 - [80] D. H. Bailey, Y. Hida, X. S. Li, B. Thompson, K. Jeyabalan, and A. Kaiser, “qd - High-Precision Software Directory,” crd-legacy.lbl.gov/~dhbailey/mpdist/, p. modified BSD license, 2000.
 - [81] D. H. Bailey, “High-precision floating-point arithmetic in scientific computation,” *Computing in Science Engineering*, vol. 7, pp. 54–61, 2005.
 - [82] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C : the art of scientific computing*. The Art of Scientific Computing, Cambridge: Cambridge University Press, first ed., 1988.
 - [83] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical recipes : the art of scientific computing*. Cambridge: Oxford University Press, third ed., 2007.
 - [84] K. Schmedders, *Numerical Optimization Methods in Economics*. London: Palgrave Macmillan UK, 2017.
 - [85] K. L. Judd, *Numerical methods in economics*. MIT press, 1998.
 - [86] S. Sra, S. Nowozin, and S. J. Wright, *Optimization for machine learning*. MIT Press, 2012.
 - [87] D. B. Fogel, “An introduction to simulated evolutionary optimization,” *IEEE trans. neural. networks.*, vol. 5, pp. 3–14, 1994.
 - [88] R. H. Byrd, G. M. Chin, W. Neveitt, and J. Nocedal, “On the use of stochastic hessian information in optimization methods for machine learning,” *SIAM J. Optim.*, vol. 21, pp. 977–995, 2011.
 - [89] J. T. Tsai, W. H. Ho, T. K. Liu, and J. H. Chou, “Improved immune algorithm for global numerical optimization and job-shop scheduling problems,” *App. Math. Comp.*, vol. 194, pp. 406–424, 2007.
 - [90] S. Huang, J. Zhou, Z. Wang, Q. Ling, and Y. Shen, “Biomedical informatics with optimization and machine learning,” *EURASIP J. Bioinform. Syst. Biol.*, vol. 4, 2017.
 - [91] M. J. Powell, *The BOBYQA algorithm for constrained optimization without derivatives*. University of Cambridge: Cambridge NA Report NA2009/06, 2009.

-
- [92] R. Fletcher, *Practical methods of optimization (2nd ed.)*. John Wiley & Sons, 1987.
 - [93] D. E. King, “Dlib-ml: A Machine Learning Toolkit,” *J. Mach. Lear. Res.*, vol. 10, pp. 1755–1758, 2009.
 - [94] A. W. King, L. Rhodes, C. Readman, and H. Cox, “Effect of nuclear motion on the critical nuclear charge for two-electron atoms,” *Phys. Rev. A.*, vol. 91, pp. 042512–1–042512–6, 2015.
 - [95] A. King, L. Rhodes, and H. Cox, “Inner and outer radial density functions in correlated two-electron systems,” *Phys. Rev. A.*, vol. 93, pp. 022509–1–022509–7, 2016.
 - [96] P. J. Davis and P. Rabinowitz, *Methods of Numerical Integration*. Academic Press, 2 ed., 1984.
 - [97] T. Hahn, “The cuba library,” *Nucl. Instrum. Meth.*, vol. 51, pp. 273–277, 2006.
 - [98] A. C. Genz and A. A. Malik, “An adaptive algorithm for numeric integration over an n-dimensional rectangular region,” *J. Comput. Appl. Math.*, vol. 6, pp. 295–302, 1980.
 - [99] Y. Hardy, K. S. Tan, W.-H. Steeb, and W.-H. Steeb, “Computer algebra with symbolic c++,” 2008.
 - [100] A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, S. B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J. K. Moore, S. Singh, T. Rathnayake, S. Vig, B. E. Granger, R. P. Muller, F. Bonazzi, H. Gupta, S. Vats, F. Johansson, F. Pedregosa, M. J. Curry, A. R. Terrel, v. Roučka, A. Saboo, I. Fernando, S. Kulal, R. Cimrman, and A. Scopatz, “SymPy: symbolic computing in python,” *PeerJ Computer Science*, vol. 3, p. e103, 2017.
 - [101] P. C. Hansen, *Rank-Deficient and Discrete ill-Posed Problems*. Siam, 1998.
 - [102] L. Kaufman, “Some Thoughts on the QZ Algorithm for Solving the Generalized Eigenvalue Problem ,” *Trans. Math. Soft.*, vol. 3, pp. 65–75, 1977.
 - [103] “cexprtk.” <https://bitbucket.org/mjdr/cexprtk/src>. Accessed: 2017-07-07.
 - [104] “Global interpreter lock.” <https://wiki.python.org/moin/GlobalInterpreterLock>. Accessed: 2018-04-02.
 - [105] R. Rajagopal, *Introduction to Microsoft Windows NT Cluster Server*. "CRC Press", 1999.
 - [106] J. A. Nelder and R. Mead, “A Simplex Method for Function Minimization,” *Comput. J.*, vol. 7, pp. 308–313, 1965.
 - [107] S. G. Johnson, “The nlopt nonlinear-optimization package.” <http://ab-initio.mit.edu/nlopt>. Last visited on 12/02/2018.
 - [108] M. J. D. Powell, “An efficient method for finding the minimum of a function of several variables without calculating derivatives,” *Comput. J.*, vol. 7, pp. 155–162, 1964.

-
- [109] H. Matthies and G. Strang, "The solution of non linear finite element equations," *Int. J. Numer. Meth.*, vol. 14, pp. 1613–1626, 1979.
- [110] S. G. Nash, "Newton-type minimization via the lanczos method," *SIAM J. Numer. Anal.*, vol. 21, pp. 770–778, 1984.
- [111] M. J. D. Powell, "Direct search algorithms for optimization calculations," *Acta. Numer.*, vol. 7, pp. 287–336, 1998.
- [112] D. Kraft, *A Software Package For Sequential Quadratic Programming*. DFLVR, 1988.
- [113] J. Z. Zhang and C.-X. Xu, "Trust region dogleg path algorithms for unconstrained minimization," *Annals of Operations Research*, vol. 87, pp. 407–418, 1999.
- [114] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer New York, 2006.
- [115] M. Born and R. Oppenheimer, "Zur Quantentheorie der Molekeln," *Ann. Phys-Berlin.*, vol. 389, pp. 457–484, 1927.
- [116] B. Sutcliffe, "To what question is the clamped-nuclei electronic potential the answer?," *Theo. Chem. Acc.*, vol. 127, pp. 121–131, 2010.
- [117] M. Baer, *Beyond Born-Oppenheimer: Electronic Nonadiabatic Coupling Terms and Conical Intersections*. Wiley, 2006.
- [118] L. Cederbaum, "The exact molecular wavefunction as a product of an electronic and a nuclear wavefunction," *J. Chem. Phys.*, vol. 138, pp. 224110–1–224110–7, 2013.
- [119] B. Sutcliffe and R. Woolley, "On the quantum theory of molecules," *J. Chem. Phys.*, vol. 137, pp. 22A544–1–22A544–8, 2012.
- [120] B. Sutcliffe, "The separation of electronic and nuclear motion in the diatomic molecule," *Theor. Chem. Acc.*, vol. 118, pp. 563–571, 2007.
- [121] G. Wu, C. Norris, H. Stewart, H. Cox, and A. Stace, "State-resolved uv photofragmentation spectrum of the metal dication complex $[\text{zn}(\text{pyridine})_4]^{2+}$," *Chem. Commun.*, pp. 4153–4155, 2008.
- [122] A. Picón, A. Jaroń-Becker, and A. Becker, "Enhancement of vibrational excitation and dissociation of H_2^+ in infrared laser pulses," *Phys. Rev. Lett.*, vol. 109, pp. 163002–1–163002–5, 2012.
- [123] Y. Nabekwa, Y. Furukawa, T. Okino, A. Eilanlou, J. Eiji, E. Takahashi, K. Yamanouchi, and K. Midorikawa, "Settling time of a vibrational wavepacket in ionization," *Nature. Comm.*, vol. 6, pp. 1–6, 2015.
- [124] E. Weisstein, "Barycentric coordinates. From MathWorld—A Wolfram Web Resource." <http://mathworld.wolfram.com/BarycentricCoordinates.html>. Last visited on 1/11/2016.
- [125] P. Porta Mana, "Notes on affine and convex spaces," *ArXiv e-prints*, 2011.

-
- [126] O. Weber, M. Ben-Chen, and C. Gotsman, "Complex barycentric coordinates with applications to planar shape deformation," *Comp. Graph. Forum.*, vol. 28, pp. 587–597, 2009.
- [127] M. Floater, "Generalized barycentric coordinates and applications," *Acta Numerica*, vol. 24, pp. 161–214, 2015.
- [128] A. Bhatia and R. Drachman, "Variational calculations of muonic-molecule energy levels," *Phys. Rev. A.*, vol. 30, pp. 2138–2140, 1984.
- [129] G. Herzberg and C. Jungen, "Rydberg series and ionization potential of the H_2 molecule," *J. Mol. Spec.*, vol. 41, pp. 425–486, 1972.
- [130] A. Balakrishnan, V. Smith, and B. Stoicheff, "Dissociation energy of the hydrogen molecule," *Phys. Rev. Lett.*, vol. 68, pp. 2749–2152, 1992.
- [131] A. Balakrishnan, V. Smith, and B. Stoicheff, "Dissociation energy of the hydrogen and deuterium molecules," *Phys. Rev. A.*, vol. 49, pp. 2460–2469, 1994.
- [132] A. Balakrishnan and B. Stoicheff, "The dissociation energy of deuterium," *J. Mol. Spectrosc.*, vol. 156, pp. 517–518, 1992.
- [133] A. Balakrishnan, M. Vallet, and B. Stoicheff, "Dissociation energy of the HD molecule," *J. Mol. Spectrosc.*, vol. 162, pp. 168–171, 1993.
- [134] A. M. Frolov, "Bound-state calculations of coulomb three-body systems," *Phys. Rev. A.*, vol. 59, pp. 4270–4280, 1999.
- [135] A. M. Frolov, "On the bound states in the muonic molecular ions," *Eur. Phys. J. D*, vol. 66, pp. 1–11, 2012.
- [136] F. Saavedra, E. Buendia, F. Gálvez, and A. Sarsa, "Precise variational calculation of some S-states of Coulomb three-body systems with two identical particles," *Eur. Phys. J. D.*, vol. 2, pp. 181–190, 1998.
- [137] P. Gill, D. O'Neill, and N. Besley, "Two-electron distribution functions and intracules," *Theor. Chem. Acc.*, vol. 109, pp. 241–250, 2003.
- [138] A. Thakkar and V. Smith, "Accurate charge densities and two-electron intracule functions for the heliumlike ions," *J. Chem. Phys.*, vol. 67, pp. 1191–1196, 1977.
- [139] W. Breunlich and P. Kammel, "Muon-catalysed fusion," *Annu. Rev. Nucl. Part. Sci.*, vol. 39, pp. 311–356, 1989.
- [140] S. Jones, "Muon-catalysed fusion revisited," *Nature*, vol. 321, pp. 127–133, 1986.
- [141] C. A. Coulson and A. H. Neilson, "Electron Correlation in the Ground State of Helium," *P. Phys. Soc.*, vol. 78, pp. 831–837, 1961.
- [142] R. J. Boyd and C. A. Coulson, "Coulomb hole in some excited states of helium," *J. Phys. B.*, vol. 6, pp. 782–793, 1973.

-
- [143] J. Pearson, P. Gill, J. Ugalde, and R. Boyd, "Can correlation bring electrons closer together?," *Mol. Phy.*, vol. 107, pp. 1089–1093, 2009.
- [144] V. W. Maslen, "The Fermi, or Exchange, Hole in Atoms," *P. Phys. Soc.*, vol. 69, pp. 734–740, 1956.
- [145] K. Wolfram and H. C. Max, *Electron Density and Hole Functions*. Wiley-VCH Verlag GmbH, 2001.
- [146] J. Slater, "A simplification of the hartree-fock method," *Phys. Rev.*, vol. 81, pp. 385–390, 1951.
- [147] P.-O. Löwdin, "Quantum theory of many-particle systems. i. physical interpretations by means of density matrices, natural spin-orbitals, and convergence problems in the method of configurational interaction," *Phys. Rev.*, vol. 97, pp. 1474–1489, 1955.
- [148] P.-O. Löwdin, "Quantum Theory of Many-Particle Systems. III. Extension of the Hartree-Fock Scheme to Include Degenerate Systems and Correlation Effects," *Phys. Rev.*, vol. 97, pp. 1509–1520, 1955.
- [149] C. Lee, W. Yang, and R. Parr, "Development of the colle-salvetti correlation-energy formula into a functional of the electron density," *Phys. Rev. B.*, vol. 37, pp. 785–789, 1988.
- [150] Y. I. and G. E. S., "A new correlation functional based on a transcorrelated hamiltonian," *J. Chem. Phys.*, vol. 118, pp. 2464–2469, 2003.
- [151] J. Sancho-García and F. Moscardó, "Usefulness of the colle-salvetti model for the treatment of the nondynamic correlation," *J. Chem. Phys.*, vol. 118, pp. 1054–1058, 2003.
- [152] R. Colle and O. Salvetti, "Approximate calculation of the correlation energy for the closed shells," *Theor. Chim. Acta.*, vol. 37, pp. 329–334, 1975.
- [153] A. W. King, A. L. Baskerville, and H. Cox, "Hartree-fock implementation using a laguerre-based wave function for the ground state and correlation energies of two-electron atoms," *Phil. Trans. Soc. A.*, vol. 376, pp. 20170153–1–20170153–16., 2018.
- [154] P. Atkins, J. De Paula, and R. Friedman, *Physical Chemistry: Quanta, Matter, and Change*. Oxford: Oxford University Press, 2 ed., 2013.
- [155] F. W. Olver, D. W. Lozier, R. F. Boisvert, and C. W. Clark, *NIST Handbook of Mathematical Functions*. New York, NY, USA: Cambridge University Press, 1st ed., 2010.
- [156] Y. Yamaguchi, "The Hartree-Fock Self-Consistent Field Method." <http://sirius.chem.vt.edu/wiki/doku.php?id=crawdad:programming:project3>. Last visited on 15/03/2018.
- [157] M. P. Barnett, F. W. Birss, and C. A. Coulson, "Electron-electron separation in molecular hydrogen," *Mol. Phy.*, vol. 1, pp. 44–47, 1958.

-
- [158] N. Moiseyev, J. Katriel, and R. Boyd, "Fermi and Coulomb correlations in the 21 S state of the helium isoelectronic sequence," *Theor. Chim. Acta.*, vol. 45, pp. 61–67, 1977.
- [159] R. F. Curl and C. A. Coulson, "Coulomb hole in the ground state of two-electron atoms," *P. Phys. Soc.*, vol. 85, p. 647, 1965.
- [160] C. J. C. Roothaan and G. A. Soukup, "Accurate and stable numerical Hartree–Fock calculations for atoms. I. The 1s2 ground state of H-, He, Li+, and Be++," *Int. J. Quant. Chem.*, vol. 15, pp. 449–462, 1979.
- [161] K. Szalewicz and H. J. Monkhorst, "On application of 0s orbitals in SCF calculations," *J. Chem. Phys.*, vol. 75, pp. 5785–5788, 1981.
- [162] T. Koga, H. Tatewaki, and A. J. Thakkar, "Numerical Hartree-Fock energies of singly charged cations and anions with N ,," *J. Chem. Phys.*, vol. 100, pp. 8140–8144, 1994.
- [163] A. Saavedra, E. Buendia, and F. Galvez, "Single-particle and electron-pair densities at the origin in the ground state of helium-like ions," *J. Phys. B.*, vol. 27, pp. 5131–5137, 1994.
- [164] T. Koga, M. Omura, H. Teruya, and A. J. Thakkar, "Improved Roothaan-Hartree-Fock wavefunctions for isoelectronic series of the atoms He to Ne," *J. Phys. B.*, vol. 28, pp. 3113–3121, 1995.
- [165] S. H. J. Gázquez, J. L., "Piecewise polynomial electronic wavefunctions.," *J. Chem. Phys.*, vol. 67, pp. 1887–1898, 1977.
- [166] T. Koga, S. Watanabe, K. Kanayama, R. Yasuda, and A. J. Thakkar, "Improved Roothaan–Hartree–Fock wave functions for atoms and ions with $N \leq 54$," *J. Chem. Phys.*, vol. 103, pp. 3000–3005, 1995.
- [167] A. M. Frolov, "Highly accurate evaluation of the singular properties for the positronium and hydrogen negative ions," *J. Phys. A.*, vol. 40, p. 6175, 2007.
- [168] Y. Bernard, P. F. Loos, and P. Gill, "Distribution of r 12 p 12 in quantum systems," *Mol. Phys.*, vol. 111, pp. 2414–2426, 2013.
- [169] C. Lee, W. Yang, and R. G. Parr, "Development of the Colle-Salvetti correlation-energy formula into a functional of the electron density," *Phys. Rev. B*, vol. 37, pp. 785–789, 1988.
- [170] King, A. W. and Rhodes, L. and Readman, C. and Cox, H., "Effect of nuclear motion on the critical nuclear charge for two-electron atoms," *Phys. Rev. A.*, vol. 91, pp. 042512–1–042512–6, 2015.
- [171] N. L. Guevara and A. V. Turbiner, "Heliumlike and lithiumlike ionic sequences: Critical charges," *Phys. Rev. A.*, vol. 84, pp. 064501–1–064501–4, 2011.
- [172] D. K. Gridnev, C. Greiner, and W. Greiner, "Stability of three unit charges: Necessary conditions," *J. Math. Phys.*, vol. 46, pp. 052104–1–052104–6, 2005.
- [173] J. Cioslowski and L. Guanghai, "Electron intracule densities and Coulomb holes from energy-derivative two-electron reduced density matrices," *J. Chem. Phys.*, vol. 109, pp. 8225–8231, 1998.

-
- [174] J. A. Pople and A. Seeger, "Self-consistent molecular orbital methods. XVIII. Constraints and stability in Hartree-Fock theory," *J. Chem. Phys.*, vol. 66, pp. 3045–2050, 1977.
- [175] D. Frye, A. Preiskorn, and E. Clementi, "The Hylleraas-CI method in molecular calculations. III. Implementation and numerical verification of a three-electron many-center theory," *J. Comp. Chem.*, vol. 12, pp. 560–564, 1991.
- [176] G. W. F. Drake, "Theoretical Energies For The N=1-State And 2-State Of The Helium Isoelectronic Sequence Up To Z=100," *Can. J. Phys.*, vol. 66, pp. 586–611, 1988.
- [177] W. Yoshihiro and T. Hiroshi, "Correlation energies for He isoelectronic sequence with Z=2-116 from four-component relativistic configuration interactions," *J. Chem. Phys.*, vol. 123, pp. 074322–1–074322–7, 2005.
- [178] H. Tatewaki and T. Noro, "Correlation energies for the He isoelectronic sequence with Z=1-120 from the third-order Douglas-Kroll Hamiltonian," *Chem. Phys. Lett.*, vol. 399, pp. 480–483, 2004.
- [179] M. A. Marques, C. A. Ullrich, F. Nogueira, A. Rubio, K. Burke, and E. K. Gross, *Excited States and Photochemistry*. Berlin, Heidelberg: Springer, Berlin, 2006.
- [180] C. Grebing, A. A. Masoudi, S. Dörscher, S. Häfner, V. Gerginov, S. Weyers, B. Lipphardt, F. Riehle, U. Sterr, and C. Lisdat, "Realization of a timescale with an accurate optical lattice clock," *Optica.*, vol. 3, pp. 563–569, 2016.
- [181] P. Elliott, F. Furche, and K. Burke, *Excited States from Time-Dependent Density Functional Theory*. John Wiley and Sons, Inc., 2009.
- [182] J. B. Foresman and A. Frisch, *Exploring Chemistry with Electronic Structure Methods, 3rd ed.* Gaussian, Inc., Wallingford, CT: Springer, Berlin, 2015.
- [183] J. M. Ugalde and R. J. Boyd, "The effect of electron correlation on one-electron properties in the 2 3S and 2 1S excited states of the helium atom," *Chem. Phys. Lett.*, vol. 114, pp. 197–200, 1985.
- [184] A. Burgers, D. Wintgen, and J. M. Rest, "Highly doubly excited S states of the helium atom," *J. Phys. B.*, vol. 28, pp. 3163–3183, 1995.
- [185] H. Nakashima, Y. Hijikata, and H. Nakatsuji, "Solving the electron and electron-nuclear schrödinger equations for the excited states of helium atom with the free iterative-complement-interaction method," *J. Chem. Phys.*, vol. 128, pp. 154108–1–154108–10, 2008.
- [186] E. Z. Liverts and N. Barnea, "S-states of helium-like ions," *Comp. Phys. Comm.*, vol. 183, pp. 844–845, 2012.
- [187] E. Weisstein, "Gauss hypergeometric function, ${}_2F_1$. From MathWorld—A Wolfram Web Resource." <http://functions.wolfram.com/HypergeometricFunctions/Hypergeometric2F1/02/01/>. Last visited on 6/4/2018.
- [188] E. A. Hylleraas and B. Undheim, "Numerische berechnung der 2s-terme von ortho- und par-helium," *Z. Physik.*, vol. 65, pp. 759–772, 1930.

-
- [189] J. K. L. MacDonald *Phys. Rev.*, vol. 43, pp. 830–833, 1933.
- [190] P. R. Surján, “Orthogonality constrained excited states peter,” *Chem. Phys. Lett.*, vol. 325, pp. 120–126, 2000.
- [191] H. Shull and P.-O. Löwdin, “Variation theorem for excited states,” *Phys. Rev.*, vol. 110, pp. 1466–1467, 1958.
- [192] T. K. Rebane and A. A. Kuzminskii, “Calculation of the stability domain for Coulomb systems,” *Optics and Spectroscopy*, vol. 93, pp. 833–836, 2002.
- [193] R. N. Hill, “Proof that the H- Ion Has Only One Bound State,” *Phys. Rev. Lett.*, vol. 38, pp. 643–646, 1977.
- [194] R. N. Hill, “Proof that the H⁻ ion has only one bound state. Details and extension to finite nuclear mass,” *J. Math. Phys.*, vol. 18, pp. 2316–2330, 1976.
- [195] B. Simon, “On the infinitude of finiteness of the number of bound states of an n-body quantum system. I,” *Helv. Phys. Acta.*, vol. 43, p. 607, 1970.
- [196] G. M. Zhislin, “On the finiteness of the discrete spectrum of the energy operator of negative atomic and molecular ions,” *Theor. Math. Phys.*, vol. 7, p. 571, 1971.
- [197] W. Hunziker, “On the asymptotic behaviour of atomic spectra near the continuum,” *Helv. Phys. Acta.*, vol. 48, p. 145, 1975.
- [198] J. Katriel, M. Puchalski, and K. Pachucki, “Binding energies of the lithium isoelectronic sequence approaching the critical charge,” *Phys. Rev. A.*, vol. 86, pp. 042508–1–042508–5, 2012.
- [199] J. D. Baker, D. E. Freund, R. N. Hill, and J. D. Morgan, “Radius of convergence and analytic behavior of the $\frac{1}{Z}$ expansion,” *Phys. Rev. A.*, vol. 41, pp. 1247–1273, 1990.
- [200] I. A. Ivanov, M. W. J. Bromley, and J. Mitroy, “Asymptotically exact expression for the energies of the $^3S^e$ Rydberg series in a two-electron system,” *Phys. Rev. A.*, vol. 66, pp. 042507–1–042507–5, 2002.
- [201] O. Osenda and P. Serra, “Excited state entanglement on a two-electron system near the ionization threshold,” *J. Phys. B.*, vol. 41, pp. 065502–1–065502–7, 2008.
- [202] Y. C. Lin and Y. K. Ho, “Quantum entanglement for two electrons in the excited states of helium-like systems,” *Can. J. Phys.*, vol. 93, pp. 646–653, 2015.
- [203] D. K. Gridnev and C. Greiner, “Proof that the Hydrogen-Antihydrogen Molecule Is Unstable,” *Phys. Rev. Lett.*, vol. 94, pp. 223402–1–223402–4, 2005.
- [204] J. G. Loeser and D. R. Herschbach, “Hylleraas-pekeris treatment of d-dimensional two-electron atoms,” *J. Chem. Phys.*, vol. 84, pp. 3882–3892, 1986.

-
- [205] A. D. McNaught and A. Wilkinson, *Atom: IUPAC. Compendium of Chemical Terminology (the "Gold Book")*. Oxford: Blackwell Scientific Publications, 2 ed., 1997.
- [206] A. D. McNaught and A. Wilkinson, *Molecule: IUPAC. Compendium of Chemical Terminology (the "Gold Book")*. Oxford: Blackwell Scientific Publications, 2 ed., 1997.
- [207] J. A. Wheeler, "Polyelectrons.," *Ann. N.Y. Acad. Sci.*, vol. 48, pp. 219–238, 1946.
- [208] E. A. Hylleraas, "Electron affinity of positronium," *Phys. Rev.*, vol. 71, pp. 491–493, 1947.
- [209] J. Mills, A. P, "Observation of the positronium negative ion," *Phys. Rev. Lett.*, vol. 46, pp. 717–720, 1981.
- [210] E. Mátyus, J. Hutter, U. Müller-Herold, and M. Reiher, "Extracting elements of molecular structure from the all-particle wave function," *J. Chem. Phys.*, vol. 135, pp. 204302–1–204302–12, 2011.
- [211] H. Primas, *Chemistry, Quantum Mechanics and Reductionism: Perspectives in Theoretical Chemistry*. "Springer-Verlag, Berlin", 1981.
- [212] H. Primas, "Emergence in exact natural sciences," *Acta. Polytech. Scand.*, vol. 91, pp. 83–98, 1998.
- [213] P. Claverie and S. Diner, "The concept of molecular structure in quantum theory: Interpretation problems," *Israel. J. Chem.*, vol. 19, pp. 54–81, 1980.
- [214] R. G. Woolley, "Quantum theory and molecular structure," *Adv. Phys.*, vol. 25, pp. 27–52, 1976.
- [215] M. Cafeiro, S. Bubin, and L. Adamowicz, "Non-Born-Oppenheimer calculations of atoms and molecules," *Phys. Chem. Chem. Phys.*, vol. 5, pp. 1491–1501, 2003.
- [216] M. J. Gaillard, D. S. Gemmell, G. Goldring, I. Levine, W. J. Pietsch, J. C. Poizat, A. J. Ratkowski, J. Remillieux, Z. Vager, and B. J. Zabransky, "Experimental determination of the structure of H_3^+ ," *Phys. Rev. A.*, vol. 17, pp. 1797–1803, 1978.
- [217] D. E. Woon and T. H. Dunning, "Gaussian basis sets for use in correlated molecular calculations. IV. Calculation of static electrical response properties," *J. Chem. Phys.*, vol. 100, pp. 2975–2988, 1994.
- [218] A. Martin, J.-M. Richard, and T. T. Wu, "Stability of three-unit-charge systems," *Phys. Rev. A.*, vol. 46, pp. 3697–3703, 1992.
- [219] J. P. Karr and L. Hilico, "Why Three-Body Physics Does Not Solve the Proton-Radius Puzzle," *Phys. Rev. Lett.*, vol. 109, pp. 103401–1–103401–5, 2012.
- [220] M. Rotenberg and J. Stein, "Use of Asymptotically Correct Wave Function for Three-Body Rayleigh-Ritz Calculations," *Phys. Rev.*, vol. 182, pp. 1–7, 1969.

-
- [221] A. M. Frolov and D. Bishop, “Theory of bound states in the Coulomb three-body system with unit charges,” *Phys. Rev. A.*, vol. 45, pp. 6236–6240, 1992.
- [222] T. K. Rebane and A. V. Filinsky, “Ground-State Energy and Stability of Three-Particle Coulomb Systems versus the Masses of the Particles Involved,” *Phys. Atom. Nuc.*, vol. 65, pp. 41–48, 2002.
- [223] G. Katz, K. Yamashita, Y. Zeiri, and R. Kosloff, “The fourier method for tri-atomic systems in the search for the optimal coordinate system,” *J. Chem. Phys.*, vol. 116, pp. 4403–4414, 2002.
- [224] R. N. Hill, “Proof that h- ion has only one bound-state - details and extension to finite nuclear mass,” *J. Math. Phys.*, vol. 18, pp. 2316–2330, 1977.
- [225] V. Korobov and J. M. Richard, “Mass-symmetry breaking in three-body ions,” *Phys. Rev. A.*, vol. 71, pp. 024502–1–024502–4, 2005.
- [226] U. M. Herold, “On the emergence of molecular structure from atomic shape in the $1/r_2$ harmonium model,” *J. Chem. Phys.*, vol. 124, pp. 014105–1–014105–5, 2006.
- [227] T. Koga and H. Matsuyama, “Correlated interelectronic angle densities of two-electron atoms in position and momentum spaces,” *Chem. Phys. Lett.*, vol. 375, pp. 565–570, 2003.
- [228] T. Koga, “Approximate formulas for the characteristics of interelectronic angle densities,” *Theor. Chem. Acc.*, vol. 112, pp. 431–434, 2004.
- [229] T. Koga and H. Matsuyama, “Inner and outer radial density functions in many-electron atoms,” *Theor. Chem. Acc.*, vol. 115, pp. 59–64, 2005.
- [230] T. Koga, H. Matsuyama, and A. J. Thakkar, “Interelectronic angles: Rounding out a geometric picture of the helium atom,” *Chem. Phys. Lett.*, vol. 512, pp. 287–289, 2011.
- [231] “Interelectronic angles: Rounding out a geometric picture of the helium atom,” *Chem. Phys. Lett.*, vol. 512, pp. 287–289, 2011.
- [232] T. Sako, J. Paldus, and G. H. F. Dierksen, “Angular correlation in He and He-like atomic ions: A manifestation of the genuine and conjugate Fermi holes,” *Phys. Rev. A.*, vol. 89, pp. 062501–1–062501–9, 2014.
- [233] E. Weisstein, “Binomial identity. From MathWorld—A Wolfram Web Resource.” <http://mathworld.wolfram.com/BinomialIdentity.html>. Last visited on 17/03/2018.
- [234] I. Gradshteyn, I. Ryzhik, and A. Jeffrey, *Table of Integrals, Series, and Products*. San Diego: Academic Press, 6 ed., 2000.

APPENDIX A



A.1 Ground state Run file

Ground state run file

```
1 #!/bin/bash
2
3 ##If we havent loaded the commands then load them
4 if [ -z ${BodyCommandsLoaded+x} ]
5 then
6     source ${HOME}/Dropbox/3Body/BashStartUpCommands
7 fi
8
9 A=`echo ""`    ##Dont change these. Instead change the ones
    ↳ below please. These are just the default blanks
10 B=`echo ""`
11 C=`echo ""`
12 K=`echo ""`
13
14 ##Load particle masses from master file (we just keep one in
    ↳ a single location for consistency)
15 source "../../path_to_particle_symbols"
16
17 m1=`echo "${P1}"`
18 m2=`echo "${P2}"`
19 m3=`echo "${P3}"`
20 if [ -z "${m1}" ]
21 then
22     m1=P1
23 fi
24 if [ -z "${m2}" ]
25 then
26     m2=P2
27 fi
```

```

28 if [ -z "${m3}" ]
29 then
30     m3=P3
31 fi
32
33 Z1=C1
34 Z2=C2
35 Z3=C3
36
37
38 MATRIXSIZE=`echo "MSIZE"`
39
40 Mode=`echo "OPT"`          #Select a mode. Possible modes are
    ↪ SP (single point), OPT (optimisation) and SCAN (scan).
    ↪ FYI it is possible to relaxed scans but this run script
    ↪ is not set up to do this.
41 MatrixType=`echo "METHOD"`      #possible values are A AC
    ↪ ABC K PEKERIS
42 if [ ${m1} == ${m2} ]
43 then
44     SYMM=`echo "SYM"`          #This should be SYM ANTISYM or
    ↪ ASYM
45 else
46     SYMM=`echo "ASYM"`        #This should be SYM ANTISYM or
    ↪ ASYM
47 fi
48 USE_RR=`echo "TRUE"`          #TRUE or FALSE (All caps)
49
50 ##converting to lowest mass units
51 source convertlowestmass
52 source CalcFormal ${m1} ${m2} ${m3} ${Z1} ${Z2} ${Z3}
53
54 ##These are the non linear variational parameters. In opt
    ↪ this are the initial values and in scan they are the
    ↪ minimum values.
55 A=`echo "1*${Eh2}"`          #beta + gamma
56 B=`echo "1*${Eh1}"`          #alpha + gamma
57 C=`echo "${A}+${B}"`          #alpha + beta

```

```

58 K=1
59
60
61 ##Scan only variables
62 A_Max=2
63 B_Max=2
64 C_Max=4
65 K_Max=3
66 A_Step=0.1
67 B_Step=0.1
68 C_Step=0.1
69 K_Step=0.1
70
71 tol=-8
72
73 sv=`echo "SVNAME.sv"` #Put in name of output eigen vector
74 arb=`echo ""` #Blank means use default precision, this is
    ↳ double unless you ask for a high tolerance. If not blank
    ↳ it will use this number of digits.
75 OptType=`echo "-bobyqa"` #Possible values are blank (use
    ↳ default), -NROpt (Use Numerical Recipes), -bfgs (Use
    ↳ Broyden-Fletcher-Goldfarb-Shanno algorithm best with
    ↳ unusual PES), -lbfgs (Similar to -bfgs ), -cg (The dlib
    ↳ variant of conjugate gradient This is the default),
    ↳ -bobyqa (The BOBYQA algorithm )
76 #Specify eigenvalue (root) to be calculated
77 Root=ROOTNUMBER
78

```

A.2 Excited state Run file

The following is the run file created to control and interface with the excited state code.

Excited state run file

```

1 # -*- coding: utf-8 -*-
2

```

```

3 """
4 This is the run file for the excited state method
5 Currently only works with atoms due to size of matrices for
  ↪ molecules
6 """
7 def inputvals():
8
9     #2014 CODATA
10    #p=1836.15267389
11    #e=1
12    #mu=206.7682826
13    #tau=3477.15
14    #d=3670.48296785
15    #t=5496.92153588
16    #He=7294.29954136
17
18    # Input masses
19    m1 = 1
20    m2 = 1
21    m3 = 'inf' # If infinity, place in quotes 'inf'
22
23    # Input charges
24    Z1 = -1.0
25    Z2 = -1.0
26    Z3 = 2.0
27
28    '''
29    |Omega |          Size          |
30    |      | Sym  | AntiSym | Asym |
31    |  1  |  3  |   1   |  4   |
32    |  3  | 13  |   7   | 20   |
33    |  4  | 22  |  13   | 35   |
34    |  8  | 95  |  70   | 165  |
35    | 10  | 161 | 125   | 286  |
36    | 15  | 444 | 372   | 816  |
37    | 21  | 1078 | 946   | 2024 |
38    | 25  | 1729 | 1547  | 3276 |
39    | 26  | 1925 | 1729  | 3654 |

```

```

40         | 30 | 2856 | 2600 | 5456 |
41         | 35 | 4389 | 4047 | 8436 |
42         | 40 | 6391 | 5950 | 12341|
43         | 45 | 8924 | 8372 | 17296|
44     '''
45     # Specify the location of the Kinetic, Potential and
46     ↪ Overlap matrices here
47     TTMat_path = "../../../path_to_file"
48     VVMat_path = "../../../path_to_file"
49     SSMat_path = "../../../path_to_file"
50
51     File_mat = 3276 # Input the matrix size stored in
52     ↪ the input matrix file
53     mat_size = 455 # Input desired matrix size
54
55     Calc = 'OPT' # Options are OPT = optimisation, SP =
56     ↪ single point, SCAN = Energy surface scan
57     Method = 'ABC' # Options are A, AB, AC, BC, ABC
58     Root = 4 # Select the eigenvalue you wish to
59     ↪ calculate/optimize. Only works with syevd and
60     ↪ ordered QZ options
61     # For triplet case, -1 from the desired root as
62     ↪ ground state for triplet is 23S
63     Prs = 4 # Number of threads to evaluate the matrix
64     ↪ on
65     EV = True # Set to True if you want the eigenvector
66     ↪ printed to file
67
68     # Hmeth, Tmeth, Vmeth and Smeth represent the method
69     ↪ of arithmetic evaluation of the Hamiltonian,
70     ↪ kinetic, potential and overlap matrices. The two
71     ↪ options are 'cexprtk' and 'eval'
72     Hmeth = 'cexprtk'
73     Tmeth = 'cexprtk'
74     Vmeth = 'cexprtk'
75     Smeth = 'cexprtk'
76
77     # Choose the Eigenvalue solver

```

```

67 EigSolver = 'syevd' # Current Options: 'syevd', 'QZ'.
    ↳ 'syevd' is a wrapper to the LAPACK syevd
    ↳ algorithm. Very fast but uses Cholesky
    ↳ decomposition so not brilliant for ill
    ↳ conditioned matrices. QZ uses the Schur
    ↳ decomposition of the LHS and RHS matrices.
68
69 # This option is if you want to order the QZ
    ↳ eigenvalues and use the root option to select
    ↳ the one you want
70 # If they are not ordered, then the 'Nearest' option
    ↳ below must be set as it will look for the value
    ↳ closest to this number
71 OrderedQZ = False # Options: True, False
72 # Find eigenvalue closest to this number, for energy
    ↳ do not include negative sign
73 Nearest = 2.17522937823679 # This option is for the
    ↳ unordered QZ algorithm. It needs to know the
    ↳ eigenvalue you want to calculate. Note, the
    ↳ starting values for the NLPs need to be chosen
    ↳ wisely.
74
75 # Choose whether or not to normalise the basis set.
    ↳ This may be useful for numerically ill
    ↳ conditioned matrices. It divides all elements on
    ↳ each row by that row's diagonal element. (i.e.
    ↳ makes all diagonal elements of matrix 1)
76 # Works best with QZ algorithm
77 Norm = False # Options: True, False
78
79 # Whether or not to bound the optimisation
    ↳ calculation
80 Bounded = False # Options: True, False
81 # 1 NLP: ((x1, y1),)          2 NLP: ((x1, y1), (x2,
    ↳ y2),)                    3 NLP: ((x1, y1), (x2,
    ↳ y2), (x3, y3))
82 bnds = ((0.000000001, 0.1), (1.5, 3.5), (1.5, 4))
83

```

```

84     A = 0.1980725614057150 # Initial A value
85     B = 2.0761968399276780 # Initial B value
86     C = 2.1351280713081766 # Initial C value
87
88     # SCAN only variables
89     AMin = 0.00
90     AMax = 1
91     BMin = 2
92     BMax = 3
93     CMin = AMin + BMin
94     CMax = AMax + BMax
95     Step = 0.1
96
97     sv_name = 'e_e_infinity_Z2_1S4_455_ABC_Ortho_OPT.sv'
98     ↪ # Input the name of the wavefunction file
99     Optimiser = 'Nelder-Mead' # Options are:
100    ↪ Nelder-Mead, Powell, COBYLA, BFGS, L-BFGS-B, CG,
101    ↪ TNC, SLSQP
102    Tol = 1e-4 # Set the tolerance for the convergence
103    ↪ of the variational parameters
104    # If Nelder-Mead this tolerance represents the
105    ↪ tolerance on the NLPs
106    # If SLSQP this tolerance represents the tolerance
107    ↪ on the energy not the NLPs
108    # If TNC this tolerance represents the tolerance on
109    ↪ the energy not the NLPs
110
111    SP\_ARB = False # Set to true if you want a single
112    ↪ point calculation at quadruple precision (32
113    ↪ digits) run using the final optimised values
114
115    Prec = 32 # If SP\_ARB = True, specify the required
116    ↪ precision of the calculation. This is SLOW so be
117    ↪ warned.
118
119    # Save input values to a dictionary

```

```

109 dict = {'m1':m1, 'm2':m2, 'm3':m3, 'z1':Z1, 'z2':Z2,
        ↪ 'z3':Z3, 'HHMat_path':HHMat_path,
        ↪ 'TTMat_path':TTMat_path,
        ↪ 'VVMat_path':VVMat_path,
        ↪ 'SSMat_path':SSMat_path, 'File_mat':File_mat,
        ↪ 'mat_size':mat_size, 'Calc':Calc,
        ↪ 'Method':Method, 'Root':Root, 'EV':EV,
        ↪ 'Tmeth':Tmeth, 'Vmeth':Vmeth, 'Smeth':Smeth,
        ↪ 'Hmeth':Hmeth, 'EigSolver':EigSolver,
        ↪ 'OrderedQZ':OrderedQZ, 'Nearest':Nearest,
        ↪ 'Norm':Norm, 'Bounded':Bounded, 'bnds':bnds,
        ↪ 'A':A, 'B':B, 'C':C, 'sv_name':sv_name,
        ↪ 'Optimiser':Optimiser, 'Tol':Tol,
        ↪ 'SP_ARB':SP_ARB, 'Prec':Prec, 'AMin':AMin,
        ↪ 'AMax':AMax, 'BMin':BMin, 'BMax':BMax,
        ↪ 'CMin':CMin, 'CMax':CMax, 'Step':Step,
        ↪ 'Prs':Prs}
110 return dict

```


APPENDIX B

B.1 Grand Central Dispatch - Apache License

Copyright 2018 Apple Licensed under the Apache License, Version 2.0 (the “License”);

you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

B.2 MPFRC++ - General Public License

MPFR C++ (MPREAL): Multiple precision floating point arithmetic library for C++. Thread-safe, cross-platform (MSVC, GCC, ICC), one-header C++ library. Supports C++ 11 features if available, C++ 03 compatible otherwise.

Thin wrapper for MPFR: <http://mpfr.org>

Project homepage: <http://www.holoborodko.com/pavel/mpfr>

Contact e-mail: pavel@holoborodko.com

Copyright ©2008-2016 Pavel Holoborodko

Contributors:

Dmitriy Gubanov, Konstantin Holoborodko, Brian Gladman, Helmut Jarausch, Fokko Beekhof, Ulrich Mutze, Heinz van Saanen, Pere Constans, Peter van Hoof, Gael Guennebaud, Tsai Chia Cheng, Alexei Zubanov, Jauhien Piatlicki, Victor Berger,

John Westwood, Petr Aleksandrov, Orion Poplawski, Charles Karney, Arash Partow, Rodney James, Jorge Leitao. Licensing: (A) MPFR C++ is under GNU General Public License (“GPL”). (B) Non-free licenses may also be purchased from the author, for users who do not want their programs protected by the GPL.

The non-free licenses are for users that wish to use MPFR C++ in their products but are unwilling to release their software under the GPL (which would require them to release source code and allow free redistribution).

Such users can purchase an unlimited-use license from the author. Contact us for more details. GNU General Public License (“GPL”) copyright permissions statement:

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. This

program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along

with this program. If not, see <<http://www.gnu.org/licenses/>>.

B.3 Doublepresso - MIT License Copyright

Copyright 1999-2018 Free Software Foundation, Inc. Contributed by the AriC and Caramba projects, INRIA.

This file is part of the GNU MPFR Library.

The GNU MPFR Library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version.

The GNU MPFR Library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General

Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with the GNU MPFR Library; see the file COPYING.LESSER. If not, see <http://www.gnu.org/licenses/> or write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301, USA.

B.4 ASMJit - Zlib License

Copyright (c) 2008-2017, Petr Kobilicek

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

B.5 Numerical Recipes Personal Single-User License

The following license terms and the Disclaimer of Warranty, below, apply to individual purchases of the Numerical Recipes Code download product, and to purchases of the Numerical Recipes Code CD-ROM.

By purchasing this disk or code download, you acquire a Numerical Recipes Personal Single-User License. This license lets you personally use Numerical Recipes code ("the code") on any number of computers, but only one computer at a time. You

are not permitted to allow anyone else to access or use the code. You may, under this license, transfer precompiled, executable applications incorporating the code to other, unlicensed, persons, providing that (i) the application is noncommercial (e.g., does not involve the selling or licensing of the application for a fee or its use in developing commercial products or services), and (ii) the application was first developed, compiled, and successfully run by you, and (iii) the code is bound into the application in such a manner that it cannot be accessed as individual routines and cannot practicably be unbound and used in other programs. That is, under this license, your application user must not be able to use Numerical Recipes code as part of a program library or “mix and match” workbench.

Businesses and organizations that purchase this disk or code download, and that thus acquire one or more Numerical Recipes Personal Single-User Licenses, may permanently assign those licenses, in the number acquired, to individual employees. Such an assignment must be made before the code is first used and, once made, it is irrevocable and can not be transferred. In many cases, business and organizations may instead wish to purchase Numerical Recipes institutional subscriptions (see below), which have more flexibility.

B.6 Lapack - Modified BSD License

Copyright (c) 1992-2017 The University of Tennessee and The University of Tennessee Research Foundation. All rights reserved.

Copyright (c) 2000-2017 The University of California Berkeley. All rights reserved.

Copyright (c) 2006-2017 The University of Colorado Denver. All rights reserved.

COPYRIGHT

Additional copyrights may follow

HEADER

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer listed in this license in the documentation and/or other materials provided with the distribution.
- Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

The copyright holders provide no reassurances that the source code provided does not infringe any patent, copyright, or any other intellectual property rights of third parties. The copyright holders disclaim any liability to any recipient for claims brought against recipient by any third party for infringement of that parties intellectual property rights.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

B.7 Boost Software License

Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

B.8 GALib - MIT License

Copyright 1995-1996 Massachusetts Institute of Technology (MIT) all rights reserved

Copyright 1996-1999 Matthew Wall (the Author) all rights reserved

GALib License Agreement

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

1. You may copy and distribute the source code and/or library/executable code for GALib in any medium provided that you conspicuously and appropriately give credit to the author and keep intact all copyright and disclaimer notices in the library.
2. Any publications of work based upon experiments that use GALib must include a suitable acknowledgement of GALib. A suggested acknowledgement is: "The

software for this work used the GALib genetic algorithm package, written by Matthew Wall at the Massachusetts Institute of Technology."

3. The author of GALib and MIT assume absolutely no responsibility for the use or misuse of GALib. In no event shall the author of GALib or MIT be liable for any damages resulting from use or performance of GALib.

B.9 Common Public License, version 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS COMMON PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT. 1. DEFINITIONS

"Contribution" means: a) in the case of the initial Contributor, the initial code and

documentation distributed under this Agreement, and

b) in the case of each subsequent Contributor:

i) changes to the Program, and

ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents " mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including

all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

- a) it complies with the terms and conditions of this Agreement; and
- b) its license agreement:
 - i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
 - ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
 - iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
 - iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

- a) it must be made available under this Agreement; and
- b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby

agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER

RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against a Contributor with respect to a patent applicable to software (including a cross-claim or counterclaim in a lawsuit), then any patent licenses granted by that Contributor to such Recipient under this Agreement shall terminate as of the date such litigation is filed. In addition, if Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in

the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. IBM is the initial Agreement Steward. IBM may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

APPENDIX C

C.1 HF two-electron integral derivation

The form for the HF two-electron integral

$$\int_0^\infty \int_0^\infty \int_{|r_1-r_2|}^{r_1+r_2} e^{-r_1-r_2} L_p(r_1) L_q(r_1) \frac{1}{r_{12}} L_u(r_2) L_v(r_2) dV, \quad (\text{C.1})$$

where dV represents the volume element

$$dV = 8\pi^2 r_1 r_2 r_{12} dr_{12} dr_2 dr_1. \quad (\text{C.2})$$

Transform from inter-particle coordinates to perimetric coordinates using the following transformations

$$\begin{aligned} r_1 &= \frac{z_2 + z_3}{2} \\ r_2 &= \frac{z_1 + z_3}{2} \\ r_{12} &= \frac{z_1 + z_2}{2} \end{aligned} \quad (\text{C.3})$$

The Jacobian of the r_i to z_i coordinate transformation is $1/4$. After the coordinate transformation the integration in equation (C.1) becomes

$$\begin{aligned} &\int_0^\infty \int_0^\infty \int_0^\infty e^{-\frac{1}{2}(z_1+z_2+2z_3)} \frac{\pi^2}{2} (z_1 + z_3)(z_2 + z_3) L_p\left(\frac{(z_2 + z_3)}{2}\right) L_q\left(\frac{(z_2 + z_3)}{2}\right) \\ &\quad \times L_u\left(\frac{(z_1 + z_3)}{2}\right) L_v\left(\frac{(z_1 + z_3)}{2}\right) dz_1 dz_2 dz_3. \end{aligned} \quad (\text{C.4})$$

Laguerre polynomials can be written using the following closed form^[55]

$$L_n(x) = \sum_{n_i=0}^n (-1)^{n_i} \binom{n}{n_i} \frac{x^{n_i}}{n_i!}. \quad (\text{C.5})$$

Equation (C.4) becomes

$$\int_0^\infty \int_0^\infty \int_0^\infty \pi^2 e^{-\frac{1}{2}(z_1+z_2+2z_3)} \sum_{p_i}^p (-1)^{p_i} \binom{p}{p_i} \frac{\left(\frac{z_2+z_3}{2}\right)^{p_i}}{p_i!} \sum_{q_i}^q (-1)^{q_i} \binom{q}{q_i} \frac{\left(\frac{z_2+z_3}{2}\right)^{q_i}}{q_i!} \sum_{u_i}^u (-1)^{u_i} \binom{u}{u_i} \frac{\left(\frac{z_1+z_3}{2}\right)^{u_i}}{u_i!} \sum_{v_i}^v (-1)^{v_i} \binom{v}{v_i} \frac{\left(\frac{z_1+z_3}{2}\right)^{v_i}}{v_i!} dz_1 dz_2 dz_3. \quad (C.6)$$

Collecting terms and simplifying

$$\int_0^\infty \int_0^\infty \int_0^\infty \pi^2 e^{-\frac{1}{2}(z_1+z_2+2z_3)} \sum_{p_i}^p \sum_{q_i}^q \sum_{u_i}^u \sum_{v_i}^v \frac{1}{2p_i!q_i!u_i!v_i!} (-1)^{p_i+q_i+u_i+v_i} \binom{p}{p_i} \binom{q}{q_i} \binom{u}{u_i} \binom{v}{v_i} \times \left(\frac{z_1+z_3}{2}\right)^{u_i+v_i} \left(\frac{z_2+z_3}{2}\right)^{p_i+q_i} (z_1+z_3)(z_2+z_3) dz_1 dz_2 dz_3. \quad (C.7)$$

Apply the binomial expansion^[233]

$$(x+y)^n = \sum_{j=0}^n \binom{n}{j} x^{n-j} y^j, \quad (C.8)$$

to the following product from equation (C.7)

$$\left(\frac{z_1+z_3}{2}\right)^{u_i+v_i} \left(\frac{z_2+z_3}{2}\right)^{p_i+q_i}. \quad (C.9)$$

Resulting in

$$\int_0^\infty \int_0^\infty \int_0^\infty \pi^2 e^{-\frac{1}{2}(z_1+z_2+2z_3)} \sum_{p_i}^p \sum_{q_i}^q \sum_{u_i}^u \sum_{v_i}^v \frac{1}{2p_i!q_i!u_i!v_i!} (-1)^{p_i+q_i+u_i+v_i} \binom{p}{p_i} \binom{q}{q_i} \binom{u}{u_i} \binom{v}{v_i} (z_1+z_3)(z_2+z_3) \sum_{a=0}^{p_i+q_i} \sum_{b=0}^{u_i+v_i} \binom{p_i+q_i}{a} \binom{u_i+v_i}{b} \left(\frac{z_2}{2}\right)^{p_i+q_i-a} \left(\frac{z_1}{2}\right)^{u_i+v_i-b} \left(\frac{z_3}{2}\right)^{a+b} dz_1 dz_2 dz_3 \quad (C.10)$$

To make the following steps easier, constant terms are factored outside of the integrals resulting in

$$\Omega \int_0^\infty \int_0^\infty \int_0^\infty e^{-\frac{1}{2}(z_1+z_2+2z_3)} (z_1+z_3)(z_2+z_3) \sum_{a=0}^{p_i+q_i} \sum_{b=0}^{u_i+v_i} \binom{p_i+q_i}{a} \binom{u_i+v_i}{b} \times \left(\frac{z_2}{2}\right)^{p_i+q_i-a} \left(\frac{z_1}{2}\right)^{u_i+v_i-b} \left(\frac{z_3}{2}\right)^{a+b} dz_1 dz_2 dz_3. \quad (C.11)$$

Ω represents the constant terms

$$\sum_{p_i}^p \sum_{q_i}^q \sum_{u_i}^u \sum_{v_i}^v \pi^2 \frac{1}{2p_i!q_i!u_i!v_i!} (-1)^{p_i+q_i+u_i+v_i} \binom{p}{p_i} \binom{q}{q_i} \binom{u}{u_i} \binom{v}{v_i}. \quad (\text{C.12})$$

Expand equation (C.11)

$$\begin{aligned} \Omega \int_0^\infty \int_0^\infty \int_0^\infty e^{-\frac{1}{2}(z_1+z_2+2z_3)} \sum_{a=0}^{p_i+q_i} \sum_{b=0}^{u_i+v_i} \binom{p_i+q_i}{a} \binom{u_i+v_i}{b} & \left(z_2^{p_i+q_i-a} z_1^{u_i+v_i-b+1} z_3^{a+b+1} \right. \\ & + z_2^{p_i+q_i-a} z_1^{u_i+v_i-b} z_3^{a+b+2} + z_1^{u_i+v_i-b+1} z_2^{p_i+q_i-a+1} z_3^{a+b} + \\ & \left. z_3^{a+b+1} z_2^{p_i+q_i-a+1} z_1^{u_i+v_i-b} \right) 2^{-p_i-q_i-u_i-v_i} dz_1 dz_2 dz_3. \end{aligned} \quad (\text{C.13})$$

The terms in equation (C.13) now satisfy the following analytical integral^[234]

$$\int_0^\infty x^n e^{-ax} dx = \Gamma(n+1) a^{-n-1} \quad (\text{C.14})$$

Applying equation (C.14) to each term in equation (C.13) results in

$$\begin{aligned} \Omega \sum_{a=0}^{p_i+q_i} \sum_{b=0}^{u_i+v_i} \binom{p_i+q_i}{a} \binom{u_i+v_i}{b} \frac{1}{2}^{(p_i+q_i-a+1)} \Gamma(p_i+q_i-a+2) \frac{1}{2}^{-(p_i+q_i-a+2)} \\ \left(\frac{1}{2} \right)^{a+b} \Gamma(a+b+1) (1)^{-(a+b+1)} \left(\frac{1}{2} \right)^{u_i+v_i-b+1} \Gamma(u_i+v_i-b+2) \frac{1}{2}^{-(u_i+v_i-b+2)} \\ + \left(\frac{1}{2} \right)^{p_i+q_i-a} \Gamma(p_i+q_i-a+1) \frac{1}{2}^{-(p_i+q_i-a+1)} \left(\frac{1}{2} \right)^{a+b+1} \Gamma(a+b+2) (1)^{-(a+b+2)} \\ \times \left(\frac{1}{2} \right)^{(u_i+v_i-b+1)} \Gamma(u_i+v_i-b+2) \frac{1}{2}^{-(u_i+v_i-b+2)} + \left(\frac{1}{2} \right)^{p_i+q_i-a+1} \Gamma(p_i+q_i-a+2) \\ \times \frac{1}{2}^{-(p_i+q_i-a+2)} \left(\frac{1}{2} \right)^{a+b+1} \Gamma(a+b+2) (1)^{-(a+b+2)} \left(\frac{1}{2} \right)^{u_i+v_i-b} \Gamma(u_i+v_i-b+1) \\ \frac{1}{2}^{-(u_i+v_i-b+1)} + \left(\frac{1}{2} \right)^{p_i+q_i-a} \Gamma(p_i+q_i-a+1) \frac{1}{2}^{-(p_i+q_i-a+1)} \left(\frac{1}{2} \right)^{a+b+2} \Gamma(a+b+3) \\ \times (1)^{-(a+b+3)} \left(\frac{1}{2} \right)^{u_i+v_i-b} \Gamma(u_i+v_i-b+1) \frac{1}{2}^{-(u_i+v_i-b+1)} \end{aligned} \quad (\text{C.15})$$

Expanding Ω gives

$$\begin{aligned}
& \sum_{p_i}^p \sum_{q_i}^q \sum_{u_i}^u \sum_{v_i}^v \pi^2 \frac{1}{2p_i!q_i!u_i!v_i!} (-1)^{p_i+q_i+u_i+v_i} \binom{p}{p_i} \binom{q}{q_i} \binom{u}{u_i} \binom{v}{v_i} \\
& \sum_{a=0}^{p_i+q_i} \sum_{b=0}^{u_i+v_i} \binom{p_i+q_i}{a} \binom{u_i+v_i}{b} \frac{1}{2}^{(p_i+q_i-a+1)} \Gamma(p_i+q_i-a+2) \frac{1}{2}^{1-(p_i+q_i-a+2)} \\
& \left(\frac{1}{2}\right)^{a+b} \Gamma(a+b+1) (1)^{-(a+b+1)} \left(\frac{1}{2}\right)^{u_i+v_i-b+1} \Gamma(u_i+v_i-b+2) \frac{1}{2}^{1-(u_i+v_i-b+2)} \\
& + \left(\frac{1}{2}\right)^{p_i+q_i-a} \Gamma(p_i+q_i-a+1) \frac{1}{2}^{1-(p_i+q_i-a+1)} \left(\frac{1}{2}\right)^{a+b+1} \Gamma(a+b+2) (1)^{-(a+b+2)} \\
& \times \left(\frac{1}{2}\right)^{(u_i+v_i-b+1)} \Gamma(u_i+v_i-b+2) \frac{1}{2}^{1-(u_i+v_i-b+2)} + \left(\frac{1}{2}\right)^{p_i+q_i-a+1} \Gamma(p_i+q_i-a+2) \\
& \times \frac{1}{2}^{1-(p_i+q_i-a+2)} \left(\frac{1}{2}\right)^{a+b+1} \Gamma(a+b+2) (1)^{-(a+b+2)} \left(\frac{1}{2}\right)^{u_i+v_i-b} \Gamma(u_i+v_i-b+1) \\
& \frac{1}{2}^{1-(u_i+v_i-b+1)} + \left(\frac{1}{2}\right)^{p_i+q_i-a} \Gamma(p_i+q_i-a+1) \frac{1}{2}^{1-(p_i+q_i-a+1)} \left(\frac{1}{2}\right)^{a+b+2} \Gamma(a+b+3) \\
& \times (1)^{-(a+b+3)} \left(\frac{1}{2}\right)^{u_i+v_i-b} \Gamma(u_i+v_i-b+1) \frac{1}{2}^{1-(u_i+v_i-b+1)}
\end{aligned} \tag{C.16}$$

This represents the analytical solution to equation (C.1). After simplifying and collecting terms

$$\begin{aligned}
& \sum_{p_i=0}^p \sum_{q_i=0}^q \sum_{u_i=0}^u \sum_{v_i=0}^v \sum_{a_i=0}^{p_i+q_i} \sum_{b_i=0}^{u_i+v_i} 2^{a_i+b_i-p_i-q_i-u_i-v_i+1} (-1)^{p_i+q_i+u_i+v_i+1} \pi^2 \left(a_i^2 - 2a_i b_i \right. \\
& \left. + b_i^2 - p^2 - 2p_i q_i - 2p_i u_i - 2p_i v_i - q_i^2 - 2q_i u_i - 2q_i v_i - u^2 - 2u_i v_i - v^2 + a_i \right. \\
& \left. + b_i - 7p_i - 7q_i - 7u_i - 7v_i - 10 \right) (p_i+q_i+u_i+v_i-a_i-b_i)! (p_i+q_i)! (u_i+v_i)! \\
& \times \frac{p!q!u!v!}{u_i!^2 v_i!^2 u_i!^2 v_i!^2 (p_i+q_i-a_i)! (q_i+u_i-b_i)! (p-p_i)! (q-q_i)! (u-u_i)! (v-v_i)!}
\end{aligned} \tag{C.17}$$

To simplify the form further

$$\begin{aligned}
& \sum_{p_i, q_i, u_i, v_i=0}^{p, q, u, v} \sum_{a_i=0}^{p_i+q_i} \sum_{b_i=0}^{u_i+v_i} 2^{a_i+b_i-\phi+1} (-1)^{\phi+1} \pi^2 \left(a_i^2 - 2a_i b_i + b_i^2 - p^2 - 2p_i q_i - 2p_i u_i - 2p_i v_i \right. \\
& \left. - q_i^2 - 2q_i u_i - 2q_i v_i - u^2 - 2u_i v_i - v^2 + a_i + b_i - 7\phi - 10 \right) (\phi - a_i - b_i)! (p_i+q_i)! (u_i+v_i)! \\
& \times \frac{p!q!u!v!}{u_i!^2 v_i!^2 u_i!^2 v_i!^2 (p_i+q_i-a_i)! (q_i+u_i-b_i)! (p-p_i)! (q-q_i)! (u-u_i)! (v-v_i)!}
\end{aligned} \tag{C.18}$$

where $\phi = p_i + q_i + u_i + v_i$.